Frontiers in Artificial Intelligence and Applications

ARTIFICIAL INTELLIGENCE IN EDUCATION

Building Technology Rich Learning Contexts that Work

Edited by
Rosemary Luckin
Kenneth R. Koedinger
Jim Greer

IOS Press

VISIT…

LANZAROTE Caliente.com

ARTIFICIAL INTELLIGENCE IN EDUCATION

Frontiers in Artificial Intelligence and Applications

FAIA covers all aspects of theoretical and applied artificial intelligence research in the form of monographs, doctoral dissertations, textbooks, handbooks and proceedings volumes. The FAIA series contains several sub-series, including "Information Modelling and Knowledge Bases" and "Knowledge-Based Intelligent Engineering Systems". It also includes the biennial ECAI, the European Conference on Artificial Intelligence, proceedings volumes, and other ECCAI – the European Coordinating Committee on Artificial Intelligence – sponsored publications. An editorial panel of internationally well-known scholars is appointed to provide a high quality selection.

Volume 158

Recently published in this series

Vol. 157. B. Goertzel and P. Wang (Eds.), Advances in Artificial General Intelligence: Concepts, Architectures and Algorithms – Proceedings of the AGI Workshop 2006
Vol. 156. R.M. Colomb, Ontology and the Semantic Web
Vol. 155. O. Vasilecas et al. (Eds.), Databases and Information Systems IV – Selected Papers from the Seventh International Baltic Conference DB&IS'2006
Vol. 154. M. Duží et al. (Eds.), Information Modelling and Knowledge Bases XVIII
Vol. 153. Y. Vogiazou, Design for Emergence – Collaborative Social Play with Online and Location-Based Media
Vol. 152. T.M. van Engers (Ed.), Legal Knowledge and Information Systems – JURIX 2006: The Nineteenth Annual Conference
Vol. 151. R. Mizoguchi et al. (Eds.), Learning by Effective Utilization of Technologies: Facilitating Intercultural Understanding
Vol. 150. B. Bennett and C. Fellbaum (Eds.), Formal Ontology in Information Systems – Proceedings of the Fourth International Conference (FOIS 2006)
Vol. 149. X.F. Zha and R.J. Howlett (Eds.), Integrated Intelligent Systems for Engineering Design
Vol. 148. K. Kersting, An Inductive Logic Programming Approach to Statistical Relational Learning
Vol. 147. H. Fujita and M. Mejri (Eds.), New Trends in Software Methodologies, Tools and Techniques – Proceedings of the fifth SoMeT_06
Vol. 146. M. Polit et al. (Eds.), Artificial Intelligence Research and Development
Vol. 145. A.J. Knobbe, Multi-Relational Data Mining

ISSN 0922-6389

Artificial Intelligence in Education

Building Technology Rich Learning Contexts That Work

Edited by

Rosemary Luckin

London Knowledge Lab, Institute of Education, University of London, UK

Kenneth R. Koedinger

Human-Computer Interaction Institute, Carnegie Mellon University, USA

and

Jim Greer

ARIES Laboratory, Department of Computer Science, University of Saskatchewan, Canada

IOS Press

Amsterdam • Berlin • Oxford • Tokyo • Washington, DC

ISBN 978-1-58603-764-2
Library of Congress Control Number: 2007928888

Publisher
IOS Press
Nieuwe Hemweg 6B
1013 BG Amsterdam
Netherlands
fax: +31 20 687 0019
e-mail: order@iospress.nl

Distributor in the UK and Ireland
Gazelle Books Services Ltd.
White Cross Mills
Hightown
Lancaster LA1 4XS
United Kingdom
fax: +44 1524 63232
e-mail: sales@gazellebooks.co.uk

Distributor in the USA and Canada
IOS Press, Inc.
4502 Rachael Manor Drive
Fairfax, VA 22032
USA
fax: +1 703 323 3668
e-mail: iosbooks@iospress.com

PRINTED IN THE NETHERLANDS

Preface

The 13th International Conference on Artificial Intelligence in Education (AIED-2007) is being held July 9–13, 2007, in Los Angeles, California. AIED Conferences are organized by the International AIED Society on a biennial basis. The goal of the International Artificial Intelligence in Education (AIED) Society is to advance knowledge and promote research and development in the field of Artificial Intelligence in Education. AIED is an interdisciplinary community at the frontiers of the fields of computer science, education and psychology. It promotes rigorous research and development of interactive and adaptive learning environments for learners of all ages, across all domains. The society brings together a community of members in the field through the organization of the AIED Conferences, a Journal, and other activities of interest. The AIED conferences are the main International forum for reporting the best international research in the field of AI in Education. The conferences provide the opportunity for the exchange of information and ideas on related research, development and applications. Previous conferences have been held in Kobe, Japan in 1997; Le Mans, France in 1999; San Antonio, USA in 2001; Sydney, Australia in 2003 and Amsterdam, The Netherlands in 2005.

Each conference adopts a theme that reflects the interests of the community and its role at the cutting edge of research. The 2007 theme is: Building Technology Rich Learning Contexts that Work. The nature of technology has changed since AIED was conceptualised as a research community and Interactive Learning Environments were initially developed. Technology is smaller, more mobile, networked, pervasive and often ubiquitous as well as being provided by the standard desktop PC. This creates the potential for technology supported learning wherever and whenever learners need and want it. However, in order to take advantage of this potential for greater flexibility we need to understand and model learners and the contexts with which they interact in a manner that enables us to design, deploy and evaluate technology to most effectively support learning across multiple locations, subjects and times. The AIED community has much to contribute to this endeavour.

Here are some statistics: Overall, we received 192 submissions for full papers and posters. 60 of these (31%) were accepted and published as full papers, and a further 52 are included here as posters. Full papers each have been allotted 8 pages in the Proceedings; posters have been allotted 3 pages. The conference also includes 2 interactive events, 10 workshops, 5 tutorials, and 16 papers in Young Researcher's Track. Each of these has been allotted a one-page abstract in the Proceedings; the workshops, tutorials, and YRT papers also have their own Proceedings, provided at the conference itself. Also in the Proceedings are brief abstracts of the talks of the four invited speakers:

Roxana Moreno, University of New Mexico
Tak-Wai Chan, National Central University of Taiwan
Danaë Stanton Fraser, University of Bath in the United Kingdom
Gregory Abowd, Georgia Institute of Technology

For the first time in the AIED conference we instituted a meta-review process. Thanks to our Senior Program Committee members, this change went quite smoothly. We believe that not only was the quality of the reviewing better, but the reviewing process was more rewarding with reviewers able to see their colleagues views and, if needed, discuss differences of opinion. Thanks, also, to the reviewers who were recruited by the Senior Program Committee members to help out in this critical task.

We would like to thank the many people who helped make the conference possible. Firstly members of Lewis Johnson's Local Organizing Committee, Jihie Kim, Andre Valente, Carole Beal and Chad Lane. Our Sponsorhsip chair Art Graesser and Rebecca Campbell, our copy editor. Special thanks to Jim Greer, our AIED Society President, who made sure we stayed on schedule. The committees organizing the other events at the conference have helped to make the conference richer and broader. Young Researcher's Track, chaired by Judith Good and Kaska Porayska-Pomsta; Tutorials, chaired by Roger Azevedo and Carolyn Rosé; and Workshops chaired by Ivon Arroyo and Joe Beck.

For those who enjoyed the contributions in this Proceedings, we recommend considering joining the International Society for Artificial Intelligence in Education: http://www.iaied.org. We certainly hope that you all enjoy the AIED-2007 conference.

Kenneth R. Koedinger, Carnegie Mellon University, USA
Rosemary Luckin, University of London, UK

Program Committee Co-Chairs

Executive Committee of the International AI-ED Society

Senior Program Committee

Esma Aimeur (Canada)
Vincent Aleven (USA)
Lora Aroyo (Netherlands)
Ivon Arroyo (USA)
Kevin Ashley (USA)
Roger Azevedo (USA)
Joe Beck (USA)
Benjamin Bell
Gautam Biswas (USA)
Bert Bredeweg (Netherlands)
Paul Brna (UK)
Peter Brusilovsky (USA)
Susan Bull (UK)
Tak-Wai Chan (Taiwan)
Yam-San Chee (Singapore)
Cristina Conati (Canada)
Ricardo Conejo (Spain)
Albert Corbett (USA)
Fiorella de Rosis (Italy)
Vania Dimitrova (UK)
Benedict du Boulay (UK)
Isabel Fernandez de Castro (Spain)
Claude Frasson (Canada)
Judith Good (UK)
Arthur Graesser (USA)
Monique Grandbastien (France)
Jim Greer (Canada)
Neil Heffernan (USA)
Ulrich Hoppe (Germany)
Mitsuru Ikeda (Japan)
Lewis Johnson (USA)
Akihiro Kashihara (Japan)
Judy Kay (Australia)
Kenneth R. Koedinger (USA)
Susanne Lajoie (Canada)
Chee-Kit Looi (Singapore)
Rosemary Luckin (UK)
Gordon McCalla (Canada)
Erica Melis (Germany)
Antonija Mitrovic (New Zealand)
Riichiro Mizoguchi (Japan)
Raphael Morales (Mexico)
Jack Mostow (USA)
Helen Pain (UK)
Ana Paiva (Portugal)
Kaska Porayska-Pomsta (UK)
Marta Rosatelli (Brazil)
Carolyn Rosé (USA)
Amy Soller (USA)
Akira Takeuchi (Japan)
Pierre Tchounikine (France)
Wouter van Joolingen (Netherlands)
Kurt VanLehn (USA)
Julita Vassileva (Canada)
Felisa Verdejo (Spain)
Peter Weimer-Hastings (USA)
Beverly Woolf (USA)
Kalina Yacef (Australia)

Additional Reviewers

Marie-Hélène Abel
Gregory Aist
Fabio Akhras
Lisa Anthony
Mohd Anwar
Ana Arruate
Eric Atwell
Ari Bader-Natal
Ryan Baker
Nelson Baloian
Tiffany Barnes
Evandro Barros Costa
Beatriz Barros
Scott Bateman
Carole Beal
Stephen Blessing
Denis Bouhineau
Jacqueline Bourdeau
Anders Bouwer
Crescencio Bravo
Christopher Brooks
Andrea Bunt
Charles Callaway
Rosa M. Carro
Alison Cawsey
Hao Cen
Stefano A. Cerri
Weiqin Chen
Sherry Chen
Patrick Chipman
Mark Core
Richard Cox
Scotty Craig
Sidney D'Mello
Declan Dagger
Ben Daniel
Nicolas Delestre
Cyrille Desmoulins
Vladan Devedzic
Darina Dicheva
Toby Dragon
Aude Dufresne
Myroslava Dzikovska
Jon A. Elorriaga
Mingyu Feng
Reva Freedman
Elena Gaudioso
Dragan Gašević
Sebastien George
Michael Glass
Zinan Guo
Eduardo Guzmán
Amanda Harris
Shinobu Hasegawa
Yusuke Hayashi
Pentti Hietala
Tsukasa Hirashima
Hilary Holz
Wong Lung Hsiang
Kim Issroff
G. Tanner Jackson
Pamela Jordan
Jelena Jovanovic
Toshinobu Kasai
Sandra Katz
Yong Se Kim
Paul Kirschner
Styliany Kleanthous
Tomoko Kojiri
Tatsuhiro Konishi
Rob Koper
Chad Lane
Fong-Lok Lee
James Lester
Jochem Liem
Chun Wai Liew
Frank Linton
Diane Litman
Tomasz Loboda
Javier Lopez
Vanda Luengo
Maxim Makatchev
Manuel Marques-Pita
Alke Martens
Brent Martin
Santosh Mathan
Noboru Matsuda
Manolis Mavrikis
Phil McCarthy
Eva Millan
Olivier Motelet
Kasia Muldner
Antoinette Muntjewerff
William Murray
Kiyoshi Nakabayashi
Jean-François Nicaud
Roger Nkambou
Toshio Okamoto
Kyparisia Papanikolaou
Fábio Paraguaçu
Darren Pearce
Jose Perez-de-la-Cruz
Diana Perez-Marin
Jean-Philippe Pernin
Niels Pinkwart
Carlos Pinto
Sowmya Ramachandran
Leena Razzaq
Genaro Rebolledo-Mendez
Christophe Reffay
Peter Reimann
Paola Rizzo
Judy Robertson
Miguel Rodriguez-Artacho
Ido Roll
Khalid Rouane
Marguerite Roy
Paulo Salles
Daniel Schwartz
Miky Sharples
Erin Shaw
Tim Smith
Sergey Sonovsky
Itana Stiubiener
Pramundi Suraweera
Jason Tan
Roger Taylor
Josephine Tchetagni
Patricia Tedesco
Monica Trella
Evangelos Tirantafillou
Joshua Underwood
Elizabeth Uruchurtu
Lucile Vadcard
Andre Valente

Conference Organizing Committee

Conference Chair: Jim Greer, University of Saskatchewan, Canada
Program Co-Chairs: Kenneth R. Koedinger, Carnegie Mellon University, USA
Rosemary Luckin, University of London, UK

Local Arrangements: Lewis Johnson, University of Southern California, USA
Publicity: Jihie Kim, University of Southern California, USA
Treasurer: Andre Valente, University of Southern California, USA
Hotel and Banquet Chairs: Ron Stevens and John Stallings, UCLA, USA
Sponsorship Chair: Art Graesser, University of Memphis, USA
Registration Chair: Carole Beal, University of Southern California, USA
Attendee Materials Chair: Erin Shaw, University of Southern California, USA
Student Volunteer Chair: Chad Lane, University of Southern California, USA

Tutorial Chairs

Roger Azevedo, University of Memphis, USA
Carolyn Rosé, Carnegie Mellon University, USA

Workshop Chairs

Ivon Arroyo, University of Massachusetts Amherst, USA
Joe Beck, University of Massachusetts Amherst, USA

Young Researchers' Track Chairs

Judith Good, University of Sussex, UK
Kaska Porayska-Pomsta, University of Edinburgh, UK

Interactive Event Chair

Hannes Högni Vilhjálmsson, Reykjavik University, Iceland

Co-Sponsored by

The International Artificial Intelligence in Education Society (www.iaied.org)
University of Southern California/Information Sciences Institute (USC/ISI)

With Support from

National Science Foundation

Institute for Intelligent Systems (IIS)
University of Memphis, USA

FedEx Institute of Technology (FIT)
University of Memphis, USA

Creative Design and Intelligent Tutoring Systems (CREDITS)
Research Center, Sungkyunkwan Univesity, Korea

Pittsburgh Science of Learning Center

Association for the Advancement of Artificial Intelligence (AAAI)

Alelo Incorporated

Contents

Representation

Motivation

Emotion and Affect

Metacognition

ITS, Feedback and Scaffolding

Learner Modelling

Linguistics and Language Technology

Linguistics and Language Technology, Tutorial Dialogues

Collaboration

Workshop Summaries

Tutorial Abstracts

Interactive Events Summaries

This page intentionally left blank

Keynote Speaker Abstracts

This page intentionally left blank

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

The Relationship Between Minds-On and Hands-On Activity in Instructional Design: Evidence from Learning with Interactive and Non-Interactive Multimedia Environments

Keynote Speaker: Roxana MORENO
University of New Mexico
http://www.unm.edu/~moreno/

What is the role of minds-on and hands-on activities on learning from multimedia environments? In this presentation, Dr. Moreno will review a set of experimental studies aimed at testing design principles that are based on cognitive and motivation theories of learning. Specifically, she will discuss whether and under what conditions including guidance, activity, reflection, feedback, and control promotes students' learning and perceptions about learning. Finally, Dr. Moreno will offer directions for future instructional technology research.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Using Automated Capture in Classrooms to Support Understanding of the Learning Experience

Keynote Speaker: Gregory ABOWD
Georgia Institute of Technology
http://www-static.cc.gatech.edu/~abowd/

In the field of ubiquitous computing, a common theme of research is to instrument an everyday environment, enabling the recording of live experiences that can be replayed at some point in the future. This general idea of "automated capture and access" has been applied in several interesting ways to educational environments. Dr. Abowd's research group first explored this concept in a project called Classroom 2000, focused on university education. More recently, he has explored classroom capture for children with developmental disabilities. This talk will summarize several case studies of classroom-based capture, with a focus on the role artificial intelligence may play to improve upon the work.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

The Technical Art of Learning Science

Keynote Speaker: Danaë STANTON FRASER
University of Bath, UK
http://staff.bath.ac.uk/pssds/

This talk focuses on the conference theme by describing techniques for creating 'Rich Learning Contexts that Work'. Dr. Stanton Fraser will discuss how newly available technologies can be appropriated for use in scientific investigations in schools. This enables a hands-on approach where traditionally the emphasis has dwelt on teaching findings over learning practice. She will describe examples of interventions in schools in which new designs for simulating science are introduced onto technologies that are independently emerging in the child's world and evaluated there. These examples highlight the possibility of sharing data across potentially large numbers of schools and between large numbers of learners who have existing access to these new technologies. This work also points to the need for retaining context to aid interpretation, and understanding the validity and appropriateness of data and method. Dr. Stanton Fraser's talk will highlight the methods that we use to engage children in the learning process and our findings from various projects where we have evaluated: logic programming using Lego; scaling up scientific investigations through the children's mobile phones; and incorporating the use of web and broadcast media. She will argue that our direct 'heavyweight' engagement methods ensure such technology-rich learning contexts will work and scale up to national size programmes.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Humanity-Based Classroom: Teaching is Caring and Learning is Joyful

Keynote Speaker: Tak-Wai CHAN
Graduate Institute of Network Learning Technology,
National Central University of Taiwan
chan@cl.ncu.edu.tw

This talk is twofold. The first part is an attempt to synthesize conceptually four seemingly diverging subfields of technology enhanced learning (TEL), including Artificial Intelligence in Education or Intelligent Tutoring Systems (AIED/ITS) and Computer Supported Collaborative Learning (CSCL) as well as two emerging subfields, namely, Mobile and Ubiquitous Learning Environments (MULE) and DIgital Game and Intelligent Toy Enhanced Learning (DIGITEL). The synthesis requires two constructs. For analyzing the trends of current TEL research and anticipating where they are leading to, we need the notion of is the first construct, namely, the three strands of TEL research: dream-based research, adoption-based research and humanity-based research. Dream-based research is to explore the potential implication of emerging technologies to learning, adoption-based research intends to prove the feasibility of spreading TEL in the real world practice; and humanity-based research tries to develop an individual's capacity both from the individual's and the social perspectives. Despite the blurring boundaries, these strands can help identify objectives and assumptions of researchers in the repertoire of TEL research.

The second construct is an extension of Self's architecture of ITS proposed in 1974. Apart from being a framework for understanding the different strengths, emphases and contributions of these subfields' in TEL as a whole, the architecture provides a structure for the conceptual synthesis. This synthesis indicates a possibility for realizing the humanity-based classroom (HBC) built on one-to-one K12 classroom in which every learner in the classroom learns with at least one wireless enabled computing device. And this is the second part of this talk. In HBC, the teacher's teaching is caring and the learners' learning is joyful. Individually, HBC aims at optimizing individual capacity development while socially, it targets at fostering a caring and humane classroom. The former stresses on development of an individual's capacity to the extent as if every learning resource in the classroom, including the fellow classmates and the teacher, is for the sake of the benefit of that individual whereas the later treats the classroom as a platform for nurturing every learner to be a good members in the class and hence, hopefully, to be able to extend this quality to the every learner's family, society, and the globe.

These two objectives, the individual capacity and the common good of communities, seem to be conflicting. Yet, they are complementary, dependent on how we define individual capacity. Only by synthesizing these four subfields both in the design and implementation in the settings of the real world practice, shall we successfully build HBC, leading to the ultimate transformation of classroom learning and hence changes of education. Although it may take many years to fulfill this goal, it is achievable by the global collective endeavor of researchers.

Acknowledgements

This talk is dedicated to John Self. The content of this talk is a product of a collective effort of a number of researchers who have been discussing with me on the topic.

Human-Computer Interface Issues

This page intentionally left blank

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

The Principle of State Expansion in Task State-Space Navigation

Darren Pearce [1] and Rosemary Luckin
London Knowledge Lab, Institute of Education

Abstract. Users typically navigate the state space of a task through the explicit manipulation of its constituent elements. This paper proposes the Principle of State Expansion (PoSE), a novel alternative to such explicit manipulation. Under the PoSE, users manipulate a *collection* of states and navigate through the state space by selecting states to expand. An abstract task is presented that contrasts the approaches and demonstrates the generality of the PoSE. This is followed by discussion of WordBird and X-Panda, two concrete implementations of software that follow the PoSE. In conclusion, the advantages over explicit manipulation are discussed especially with reference to the potential for sophisticated learner modelling.

Keywords. state-space navigation; task interface design, novel interfaces, HCI, interaction design, artificial intelligence

1. Introduction

The paper focuses on the ways in which it is possible for a user to navigate the state space of a task. For example, imagine software that presents the user with a list of words that they must assign to either adjective or noun categories. The state of this task at any particular time can be represented by the current list of adjectives and the current list of nouns. The set of all the possible different groupings of the words is the *state space* of the task. These groupings change as the user progresses and words are moved from one group to another. The user thus navigates their way through the space of possible states (the state space).

This imaginary piece of software, as described, allows *explicit manipulation* of the task state in that it allows the user to move from one state to another by picking an element and moving it *explicitly.* This paper describes the Principle of State Expansion (PoSE), a novel alternative to such explicit manipulation.

Section 2 illustrates explicit manipulation using an imaginary task. Section 3 introduces the Principle of State Expansion (PoSE) and contrasts this with explicit manipulation. Section 4 and Section 5 describe WordBird and X-Panda, two pieces of software which implement tasks that follow the PoSE. These three examples are then used in the discussion in Section 6.

2. Explicit Manipulation

Consider a task in which the user is given a row of three cells. They are then told that they will be given three letters and asked to arrange them within the cells so that they form a valid (three-letter) word. These letters will be given to them *one at a time* and, at each stage, the user should position the letters they have received so far with specific (eventual) three-letter words in mind. Figure 1 illustrates a scenario featuring a concrete instance of this task in which the user is given the three letters r, a and c in that order.

Information from this scenario can be used to annotate the state space of this task as shown in Figure 2a. Each node in this diagram is a task state such as those shown in Figure 1 and is annotated with an indication as to whether it was seen or not (within the scenario). A node can also be a solution state. Figure 2b shows the key for the annotations.[2]

[1] Correspondence to: Darren Pearce, London Knowledge Lab, 23–29 Emerald Street, London, WC1N 3QS, email: `d.pearce@ioe.ac.uk`

[2] This state-space is only for the specific instance of adding the three letters r, a and c in that order. Furthermore, the diagram only shows links for deleting or moving the most-recently-added letter.

The user is presented with a row of three blank cells.

They place the *r* thinking that the eventual word might be *red* or *rib*.

They then place the *a* thinking that the eventual word might now be *ram* or *raw*.

Once they see the *c*, they move the *r* ...

... so that they can place the *c* to form their solution.

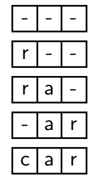
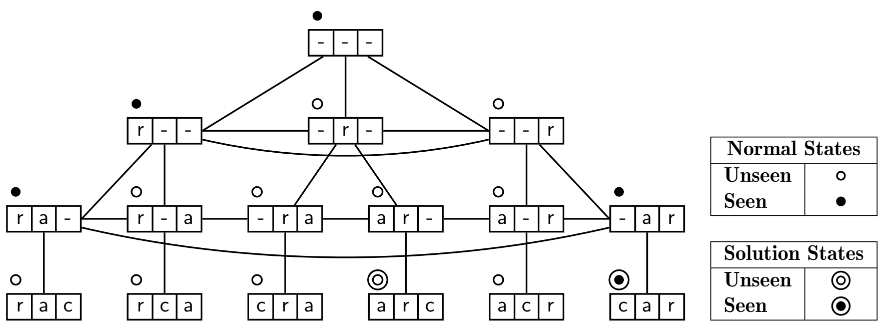


Figure 1.: Three-letter task: explicit manipulation scenario.



(a) (b)

Figure 2.: Three-letter task: state space annotated with explicit manipulation scenario information

3. The Principle of State Expansion

This section describes the Principle of State Expansion (PoSE), a novel alternative to explicit manipulation for navigating the state space of a task. Under the PoSE, the user does *not* explicitly manipulate the task state. Instead, at any particular point in time during the task, the user has a collection of an arbitrary number of task states. They navigate their way through the state space by selecting states to expand. The PoSE can therefore be seen as a type of *implicit* manipulation of the task state. Figure 3 shows another scenario for the three-letter task, this time following the PoSE.

The starting state.

The starting state is 'expanded' with the *r* leading to three new states.

The user discards *-r-* since they cannot think of any words with this pattern.

Their remaining two states are expanded with the *a*.

They discard *r-a* since no words fit this pattern.

The remaining three states are each expanded with the *c*.

The user discards all but *car*, the solution state.

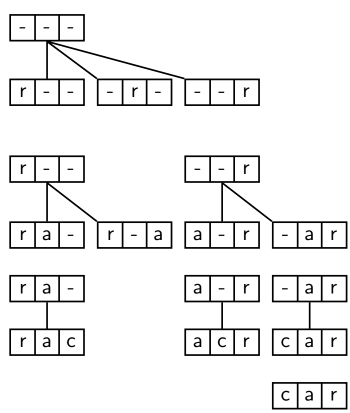


Figure 3.: Three-letter task: PoSE manipulation scenario

The state-space diagram can now be annotated with much richer information as shown in Figure 4a. The notion of a seen state is now ambiguous since it may be a state that the user saw and decided to

keep or the user saw and decided to discard. The annotations are therefore elaborated appropriately as shown in Figure 4b.

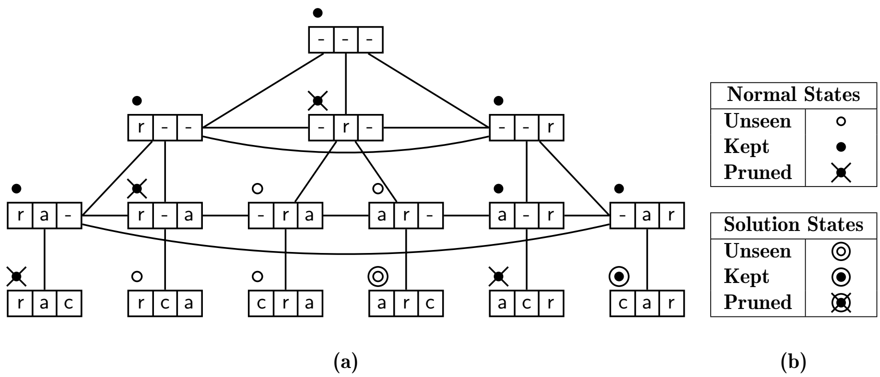


Figure 4.: Three letter task: state-space annotated with PoSE manipulation scenario information.

4. WordBird

The Riddles Project[3] developed several pieces of software designed to engage 7–11-year old children in thinking about words and word meaning since this has been shown to lead to a general increase in reading comprehension ability. One of these pieces of software was WordBird which presents the user with vague and ambiguous short stories *sentence by sentence*. The user is asked to build a graphical representation of what they feel the story might represent so far. The interface follows the PoSE, adopting a 'machine' metaphor. Figure 5 shows the interface and explains the functionality of the four slots present within the interface.

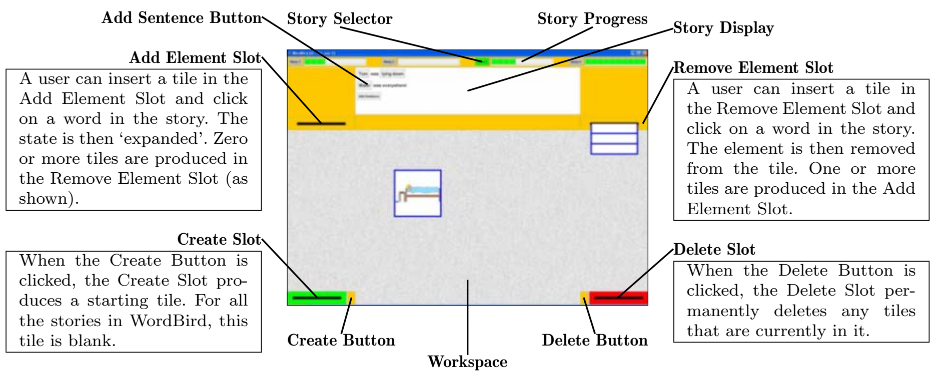


Figure 5.: The WordBird Interface.

Figure 6 presents a scenario for one particular story within the software. Note that in contrast to the three-letter task, the user is able to *choose* which task states (tiles) are expanded; the system does not expand all tiles automatically. As before, the information from this scenario can be used to annotate a state-space diagram as shown in Figure 7.[4]

[3] `http://www.riddles.sussex.ac.uk/`, EPSRC grant number: GR/R96538/01.

[4] Note that not all states in the penultimate row can have *silly boy* added to them since one of the constraints for states within this particular story is that either boy, if standing, must always be on the far left of the picture. This cannot apply to both boys at the same time.

The user inserts a starting tile into the Add Element Slot and clicks on the word *Tom* in the first sentence.

This results in a single tile consisting of Tom. The user inserts this tile into the Add Element Slot. This time they click on the phrase *lying down*.

As a result, the user receives *four* tiles each of which now consists of both Tom and a 'way in which to lie down'.

Using the Delete Slot, the user discards the tiles in which Tom is standing since Tom is not lying down in these tiles. They add *water* to the tile of Tom in bed.

The machine produces two further tiles, one with a glass of water standing on the bed-side table and the other with the glass of water spilt.

They discard the tile where the glass of water is not spilt since water is not everywhere. They try adding *water* to the tile with Tom on a deck chair.

As before they receive tiles with a glass of water spilt and unspilt. However, the machine also produces one with a swimming pool adjacent to the deck chair.

The user discards the tile with unspilt water once again and adds the next sentence. Although this sentence does not have any clickable words or phrases ...

... the user is able to infer that the tile with Tom in bed is unlikely to be a/the solution since if one is in bed, one is generally not having a day out.

Having seen the final sentence, the user adds *silly boy* to the swimming pool picture and obtains the solution tile.

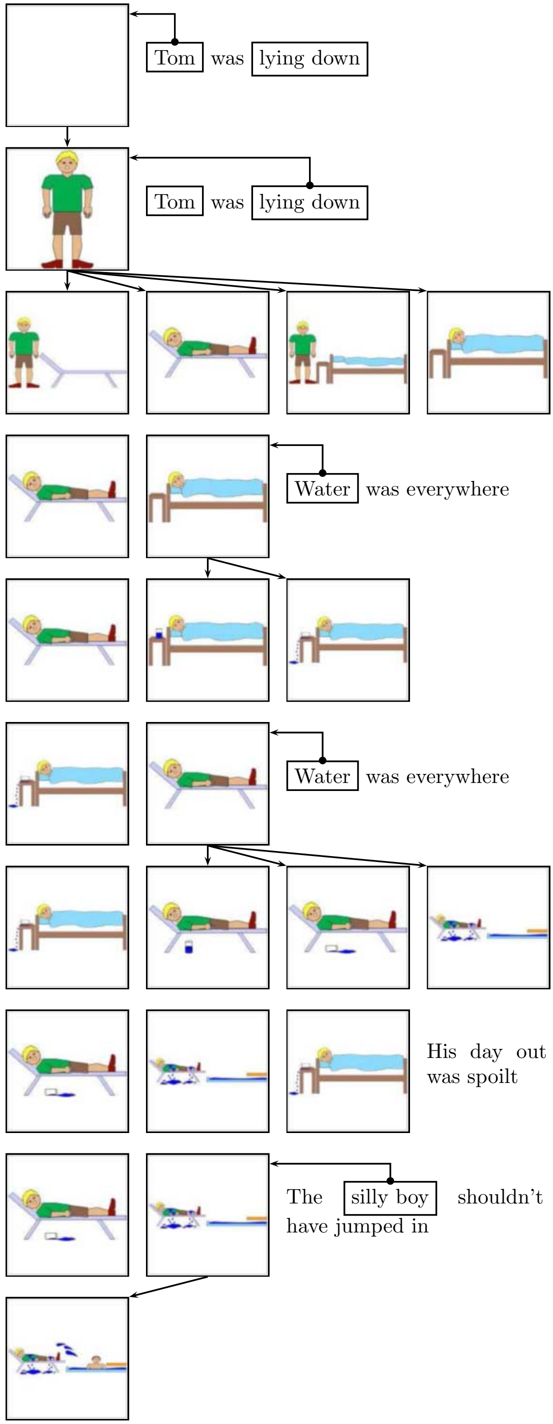


Figure 6.: WordBird scenario diagram. Each of the four sentences is shown to the right of the tile sets where appropriate. Expansion steps are indicated by arrowed lines between tiles.

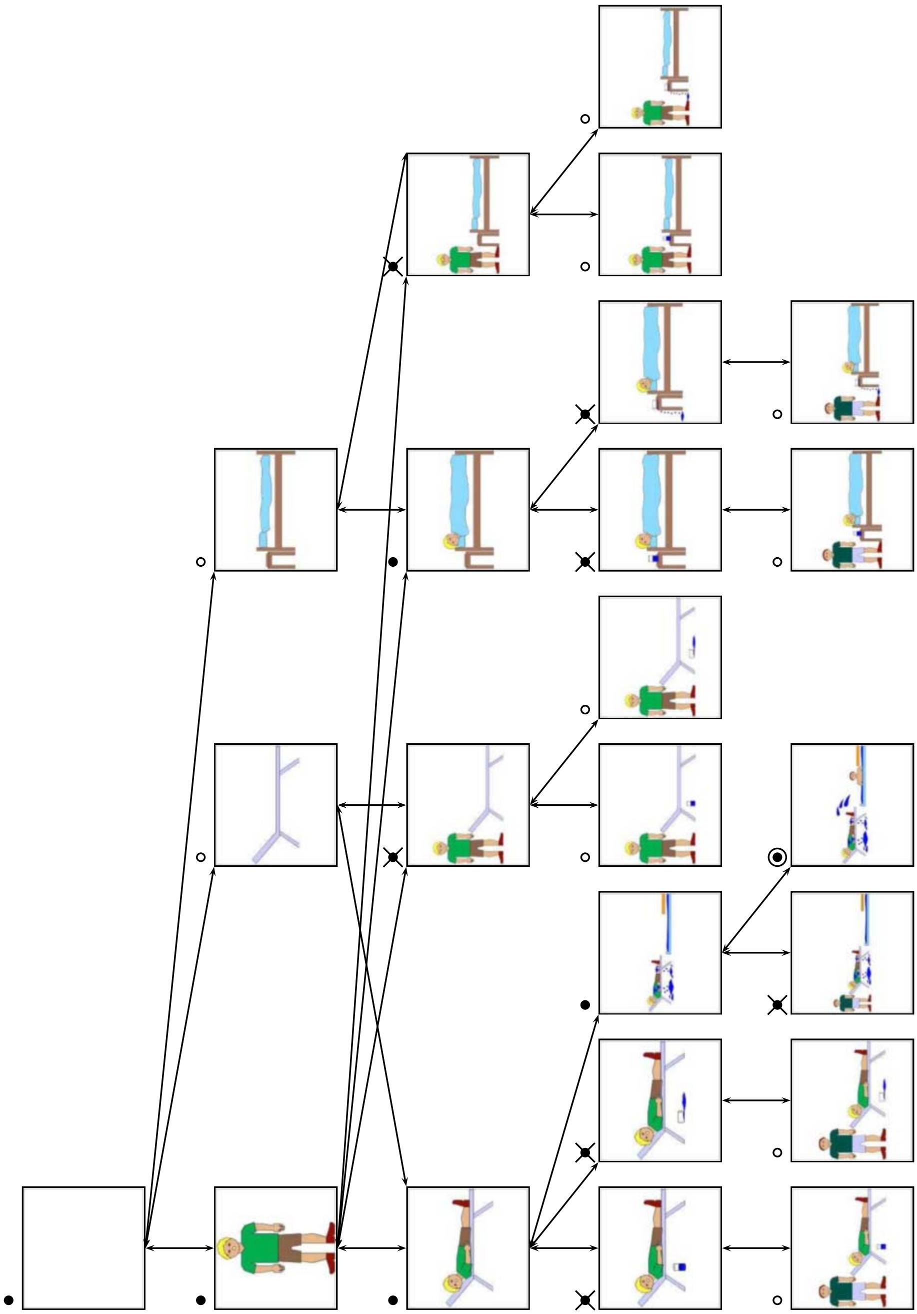


Figure 7.: WordBird state-space diagram annotated with scenario information.

5. X-Panda

X-Panda is a generalised version of WordBird and was developed as part of the the EPSRC-funded Homework Project[5] which designed a system to bridge the home and school contexts. X-Panda featured as part of the home context and provided activities for Key Stage 1 numeracy. The interface of X-Panda was in essence identical to that of WordBird, utilising a machine metaphor to implement the PoSE. The task requires the construction of two groups of three numbers where the numbers in each group follow the same rule. For example, a group might consist of 5, 10 and 20 with a rule of 'multiples of five' or it might consist of 4, 5 and 6 with a rule of 'they are in sequence'.

As with WordBird, the numbers that the children were able to add were provided one at a time so that they could incrementally build their tiles, refining and revising their hypotheses along the way. In contrast to WordBird, however, the starting tile was not blank but consisted of one number in each group as a way to narrow down the state space of the task. Figure 8 presents a scenario[6] for one particular instance of this task and this information is then used to annotate the state-space diagram shown in Figure 9.[7]

The user clicks on the Create Button and receives a start state. This consists of a single number in each group. The user adds the first element, 6, to the starting tile.

The user thinks that the first new state could be multiples of 3 and 4 and the second could be 1, 2, 3 and 4, 5, 6. They expose the next element, a 5, and use it to expand the first of these tiles.

They decide that neither of these new tiles makes sense as number groupings so ...

... they use the Delete Slot to remove both of them. This leaves a single tile which they then expand with the 5.

At this point, the user thinks that the first tile could be odds and evens. They expose 1, the next element and add it to this first tile.

This leads to two new tiles, the first of which is consistent with their odd/even hypothesis, the second of which is not.

In view of this, they discard the second tile. They then expose the final element which is a 2 and add this to their tile.

Discarding the other tile, they are left with a solution which confirms their earlier hypothesis.

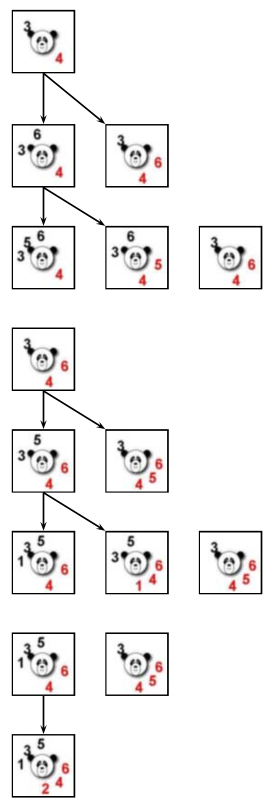


Figure 8.: X-Panda scenario diagram

[5]Grant number: RES-328-25-0027

[6]Note that each tile within this task consists of numbers arranged around a central (panda) icon. One group is those in the top-left quadrant of the circle; the other is in the bottom-right quadrant. This design was chosen so as to prevent erroneous association *between* groups; association is only *within* groups.

[7]The state-space diagram only shows links representing movement of the most-recently-added element. In practice, the children were able to move any element they had added to the tile. This was possible by inserting a tile in the 'Add' Element Slot and clicking on a number already on the tile.

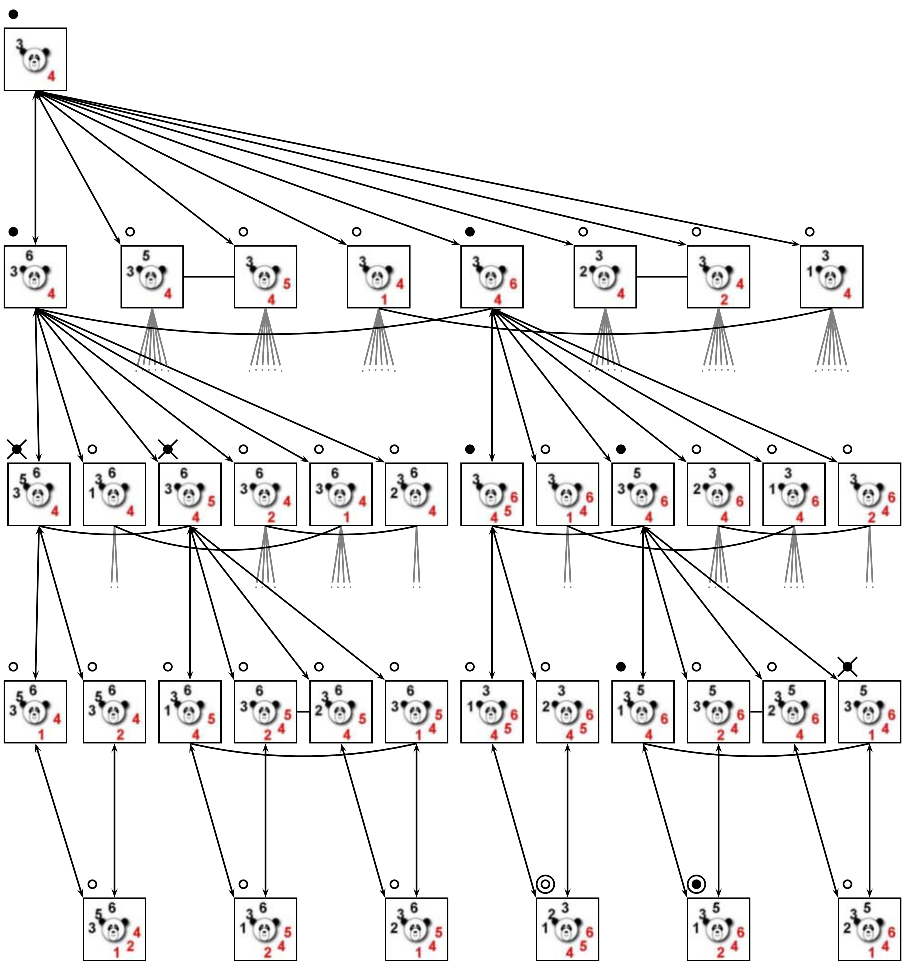


Figure 9.: X-Panda State-Space Diagram

6. Discussion

This paper has proposed the Principle of State Expansion (PoSE) as an alternative to explicit manipulation for navigating the state space of a task. It is generally applicable to any task with a well-formed state space and particularly well-suited to those tasks where elements are provided to the users one by one. This enables them to engage in the simultaneous consideration of multiple hypotheses, refining and revising these hypotheses as they progress through the state space. In contrast to explicit manipulation, tasks adopting the PoSE yield information not only on which states the user has visited but also which states the user has *explicitly* discarded and, crucially, *when* they discarded them.

This information holds significant potential for rich learner modelling. An overlay modelling approach [1,2,3] would, for example, enable the system to target support to the leading edge of the current activity of the learner. It would also be a useful way to develop open, active and collaborative learner

models [4,5,6] and to support processes such as metacognition [7,8]. It enables learners to represent, share and manipulate multiple hypotheses building upon work such as that completed by [9] and [10].

The advantages of the PoSE are even more important in a collaborative setting. Both WordBird and X-Panda do in fact also have a collaborative mode which provides two users each with their *own* set of task tiles that they alone can manipulate. This follows principles from SCOSS [11] and, more generally, the Task Sharing Framework [12]. Using PoSE in combination with these paradigms, it is possible for both users to explore completely different parts of the state space if they so desire without dominating each other's behaviour which has the potential to lead to high-quality collaboration.

The PoSE is a novel approach to task interface design and, in view of this, the design of both WordBird and X-Panda allowed the user complete freedom in the organisation of their workspace; they could create as many tiles as they wanted and lay them out within the workspace as they desired. In studies with these pieces of software, many users took advantage of this facility to explore the state space of the task as they saw fit.[8] Using video and program log data, it will be possible to investigate these various exploration strategies and its effects on the progress of the user(s) through the task. This is one of several ways in which this work is being extended. Current research also includes developing effective interfaces other than the machine metaphor for implementing the PoSE and exploring both the range of tasks for which the PoSE can be usefully applied and its potential for enriching collaborative experiences.

References

[1] P. Goldstein, "The genetic graph: a representation for the evolution of procedural knowledge," in *Intelligent Tutoring Systems* (D. Sleeman and J. S. Brown, eds.), New York: Academic Press, 1982.

[2] B. Brecht, "Student models: the genetic graph approach," *International Journal of Man-Machine Studies*, vol. 28, pp. 483–504, 1988.

[3] R. Luckin and B. du Boulay, "Designing a zone of proximal adjustment," *International Journal of Artificial Intelligence and Education*, vol. 10, no. 2, pp. 198–220, 1999.

[4] S. Bull and A. Mabbott, "20000 inspections of a domain-independent open learner model with individual and comparison views," in *Intelligent Tutoring Systems: 8th International Conference* (M. Ikeda, K. Ashley, and T.-W. Chan, eds.), (Heidelberg, Berlin), pp. 422–432, Springer-Verlag, 2006.

[5] S. Bull, P. Brna, and H. Pain, "Extending the scope of the student model," *User Modelling and User Adapted Interaction*, vol. 5, no. 1, pp. 45–65, 1995.

[6] J. Kay, "Learner control, user modeling and user-adapted interaction," *Tenth Anniversary Special Issue*, vol. 11, no. 1-2, pp. 111–127, 2001.

[7] S. J. Derry, "Metacognitive models of learning and instructional systems design," in *Foundations and Frontiers in Instructional Computing Systems* (P. H. Winne and M. Jones, eds.), (Berlin), Springer-Verlag, 1991.

[8] R. Luckin and L. Hammerton, "Getting to know me: helping learners understand their own learning needs through metacognitive scaffolding," in *Intelligent Tutoring Systems* (S. A. Cerri, G. Gouarderes, and F. Paranguaca, eds.), pp. 759–771, Berlin: Springer-Verlag, 2002.

[9] E. Toth, D. Suthers, and A. Lesgold, "Mapping to know: The effects of evidence maps and reflective assessment on scientific inquiry skills," *Science Education*, vol. 86, no. 2, pp. 264–286, 2002.

[10] T. Murray, L. Winship, R. Bellin, and M. Cornell, "Toward glass box educational simulations: Reifying models for inspection and design," in *AI-ED 2001 Workshop on External Representations in AIED: Multiple Forms and Multiple Roles*, (San Antonio, Texas), May 2001.

[11] L. Kerawalla, D. Pearce, N. Yuill, R. Luckin, and A. Harris, "I'm keeping those there, are you? The role of a new user interface paradigm — Separate Control Of Shared Space (SCOSS) — in the collaborative decision-making process." Computers and Education, forthcoming.

[12] D. Pearce, L. Kerawalla, R. Luckin, N. Yuill, and A. Harris, "The Task Sharing Framework: A generic approach to scaffolding collaboration and meta-collaboration in educational software," in *Proceedings of the 13th International Conference on Computers in Education*, (Singapore), 2005.

[8] The formal evaluation of this user interaction data will be presented in future publications; it is outside the focus of this paper

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Generating Responses to Formally Flawed Problem-Solving Statements

Helmut Horacek [a] and Magdalena Wolska [b]
[a] *Fachrichtung Informatik*
[b] *Fachrichtung Allgemeine Linguistik (Computerlinguistik)*
Universität des Saalandes, Postfach 15 11 50, D-66041 Saarbrücken, Germany
{horacek@cs.uni-sb.de,magda@coli.uni-sb.de}

Abstract. Teaching problem-solving in formal domains aims at two purposes: 1) increasing the students' skill in addressing the problem in a goal-oriented way, and 2) increasing their competence in expressing themselves formally. In a dialog, suitable tutoring strategies addressing both issues may be quite delicate, especially when meeting formally inaccurate or even faulty statements which nevertheless are useful from the problem-solving perspective. Based on evidence from Wizard-of-Oz studies on human tutoring in mathematical theorem proving, we have developed a model for addressing formally inaccurate statements within a problem-solving context. Based on an error recognition and correction module, the model performs a conceptually-motivated error categorization and generates a suitable response.

1. Introduction

Our goal in the DIALOG project is to build a prototype dialog-enabled system for tutoring mathematical proofs [1].[1] Teaching problem-solving skills in formal domains is among the most ambitious goals of tutoring systems. This task is inherently difficult because responding to students' statements in a dialog means meeting two complementary purposes: increasing the students' skills in addressing the problem in a goal-oriented way as well as their competence in expressing themselves in a formal way. Consequently, suitable tutoring strategies may be quite delicate, especially in case of formally inaccurate or faulty statements which nevertheless are useful from the problem-solving perspective.

Based on evidence from observing human tutors in Wizard-of-Oz studies on tutoring mathematical theorem proving (see Section 2), we have developed a model for addressing student statements that are formally flawed, but otherwise useful in advancing toward the solution. Based on an error recognition and correction module, the model performs a conceptually-motivated error categorization and generates a suitable response.

This paper is organized as follows. We first present empirical data on students' problem-solving behavior in two mathematical subdomains. Then we briefly present our error recognition and correction procedure. We follow with the generation of situation-adequate tutor responses. Finally, we describe a preliminary evaluation of our model and discuss other related approaches.

[1] The DIALOG project is part of the Collaborative Research Centre on *Resource-Adaptive Cognitive Processes* (SFB 378) at University of the Saarland: http://www.coli.uni-sb.de/sfb378/.

2. The Data

Our modelling is informed by analysis of two tutorial dialog corpora collected in Wizard-of-Oz experiments in which a human tutor–wizard was simulating the behavior of a system: Corpus-I [2] and Corpus-II [3]. Below, we briefly summarise the setup of the two experiments and our data.

2.1. Corpus Collection

The subjects participating in the experiments were told that they were interacting with a conversational tutoring system. They were using natural language (German) typed on the keyboard and mathematical symbols on the Graphical User Interface (GUI). The subjects and the wizards were unconstrained in their language production. The resulting dialog utterances were formulated in the typical mixed-language of mathematical discourse: natural language with interleaved mathematical expressions [5]. Overall, the collected corpora consist of 22 and 37 dialog logfiles for Corpus-I and Corpus-II respectively.

Corpus-I In the first experiment, the subjects solved three proofs in naive set theory. They were divided into three groups and tutored with *minimal feedback* (only correctness and completeness evaluation), *didactic* (the expected realization of the given proof-step was included in tutor's feedback), or *socratic* strategy (the tutor guided the student at the solution using hints, following a pre-defined hinting algorithm).

Corpus-II The subjects were solving four proofs in binary relations. The GUI provided alternative methods of inserting mathematical expressions: buttons as well as English and German Latex-like commands. The tutors were not given any tutoring algorithm, but, instead, general instructions on *socratic* tutoring. The manipulated factor was the presentation of the study-material: verbose (using a mixture of natural language and formulas) vs. formal (using mainly formulas). Further details of the setup can be found in [6]. [4] presents the results of the study on the influence of the presentation format.

2.2. Errors in Mathematical Expressions

In [7], we presented a formula error categorization motivated by errors found in student utterances in Corpus-I. Based on the analysis of Corpus-II, in which the mathematical expressions are structurally more complex, we extended our categorization to include phenomena previously unaccounted for.

On the one hand, we distinguish between error types depending on whether and, if so, how a given expression can be analyzed (parsed), and whether a domain reasoner[2] evaluates the expression as correct/wrong: a *structural error* means that on the basis of the defined constructors an analysis tree cannot be built (Category 3), a *type error* means that an expression can be successfully parsed, however, a type mismatch is identified (Category 2), finally a *logical error* is diagnosed if a correctly analyzed expression is evaluated as a false statement (Category 1).

On the other hand, an analysis of students' conceptual errors that lead to the above error types provides a different view on the error categorization: conceptual errors may be related to *operators*, *operands* (operator's arguments), or *separators* (or punctuation).

[2] To check validity of possibly ambiguous proof-step interpretations and consistency within the proof context, we use a mathematical assistant system, e.g. SCUNAK [8] or ΩMEGA [9].

	Example Formula	*Error Category*
(1)	R∪S = {x|x∈R ∧x∈S}	1
(2a)	(R ∘ S) $^{-1}$ = { (x,y) | ∃ z(z ∈ M ∧ (y,z) ∈ R $^{-1}$ ∧ (z,x) ∈ S $^{-1}$) } ⊆ S $^{-1}$ ∘ R $^{-1}$	1
(2b)	$P((A \cap B) \cup C) = P(A \cap B) \cup P(C)$	1
(3a)	(T $^{-1}$ ∘S $^{-1}$) $^{-1}$ ∪(T $^{-1}$ ∘R $^{-1}$) $^{-1}$ ⇔(y,x)∈(T $^{-1}$ ∘S $^{-1}$)∨(y,x)∈(T $^{-1}$ ∘R $^{-1}$)	2
(3b)	$(x \in b) \notin A \qquad x \subseteq K(A)$	2
(4)	(R∘S) $^{-1}$ = S $^{-1}$ ∘ R $^{-1}$ ⇒ (a,b) ∈(T $^{-1}$ ∘ R $^{-1}$) $^{-1}$ ∪ (T $^{-1}$ ∘ S $^{-1}$)	2
(5)	∃z∈M:(x,y)∈R∘T ∨(x,y)∨S∘T	1,2
(6)	(R ∪ S) ∘ T = {(x,y)| ∃ z(z ∈ M ∧ (x,z) ∈ {x|x ∈ R ∨ x ∈ S} ∧ (z,y) ∈ T)}	2
(7)	*[M:set]* . . . (x,y) ∈ M	2
(8)	S $^{-1}$ ∘ R $^{-1}$ ={(x,y)| ∃ z(z ∈ M ∧ (x,) ∈ S $^{-1}$ ∧ (z,y) ∈ R $^{-1}$)}	3
(9)	(s,r)∈(R ∘ S) $^{-1}$ . . . (s,r)∈R ∘ S	1
(10)	*[(b,a) ∈ (R ∘S), z ∈ M]* (x,z)∈R and (z,y)∈ S	1
(11)	∃z∈M: ((x,z) ∈R ∧ (z,y)∈T)∨((x,z)∈S∧(z,y)∈T	3
(12a)	(a,b) ∈ R ∘ T ∩ S ∘ T	3
(12b)	$P((A \cup C) \cap (B \cup C)) = PC \cup (A \cap B)$	3

Table 1. Examples of flawed formulas from the corpora

Table 1 shows examples of flawed formulas from our corpora.[3] Examples (1–5) concern operator errors, examples (6–10) operand (argument) errors, and (11–12) segmentation errors (i.e. errors related to use of parentheses). The error type (Categories 1–3) associated with each example is shown in the right column. Table 2 shows the more fine-grained error categorization motivated by the new phenomena, as well as associations of students' conceptual errors (error source; middle column) with the resulting error type (right-hand column). Let us take a closer look at the errors:

(1) and (2) are examples of logical errors: the expressions are well-formed, however, in (1), the logical disjunction ($\vee$) is needed, in (2a–b), a stronger or weaker assertion is expected (about equality rather than inclusion or vice-versa). In (3a), an argument type mismatch results from the use of a wrong operator for the definition, while in (3b), examples of commonly confused relations of subset and membership are shown. In (4), the expression on the right-hand side of the implication misses the inverse operator. And, finally, in (5) unrelated operators were confused: $\vee$ in place of $\in$.

Examples (6–10) show errors resulting from wrong use of operands. In (6), the same variable (x) is used in two different contexts: once as an element of a pair, and second, as a variable denoting an element of a set. In (7), a set M is declared in the task definition, however, the student appears to think that M contains pairs (i.e. is a relation). In (8), the second element of the pair is missing, resulting in a structural error. In (9), a logical error is caused by variables having been swapped. Finally, in (10), undeclared variables (x and y) are used even though a prior declaration was made for the given context.

The last block of examples concerns segmentation problems resulting from inappropriate use of parentheses (or lack thereof). In (11), the expression is incomplete (closing bracket missing). In (12a), the expression is ambiguous, and in (12b), not only a space between the operator symbol P and identifier C, but also parentheses are missing.

Let us now turn to the processing of mathematical expressions and error diagnosis and correction.

[3]Where relevant, we added prior context in *italics*.

Category	*Source*	*Error Category*
operator	dual operator (converse relation)	1
	stronger/weaker operator (inaccurate relation)	1
	commonly confused operators	2
	missing/extra operator	1,2
	confused unrelated operators	1,2
operand	ambiguous identifier use	1,2
	exchanged variable	1
	incompatible sub-structure	2
	missing operand	3
	undeclared variable (unknown identifier)	1,2
separator	incomplete expression	3
	ambiguous expression	3

Table 2. Error categories

3. Mathematical Expression Analysis and Correction

Mathematical expression processing is part of the parsing component in an architecture for processing informal mathematical discourse [10]. Processing consists of two parts: a mildly fault-tolerant analysis procedure and a correction procedure.

Mathematical expression analysis consists of three stages: (1) *detection:* mathematical expressions are identified within word-tokenized text; (2) *syntactic verification:* the identified sequence is verified as to syntactic validity and, in case of a parentheses mismatch, a parenthesis correction procedure is invoked, and (3) *parsing:* a tree representation of the expression is built by the parser. At all stages, the analysis modules have access to a list of operation and identifier symbols (variables) relevant in the given context.

Identification of mathematical expressions is based on simple indicators: single character tokens (including parentheses), multiple-character tokens consisting only of known relevant characters, mathematical symbol tokens (unicode numbers), and new-line characters. Multiple-character candidate tokens are further segmented into operators and identifiers by inserting the missing spaces. Once a candidate string is identified, "chains" of formulas are separated into individual segments.

Next, we address structural segmentation errors (Category 3) if present. Missing parentheses are inserted while observing the following preferences: (i) provide parentheses for operators that require bracketed arguments, (ii) avoid redundant parentheses (i.e. double parentheses around the same substring).

Correctly bracketed sequences are parsed by a tree-building algorithm that has access to standard required information such as: list of operators and operator precedence. The output of the parser is a set of formula trees with nodes marked as to type compatibility (Category 2) and bracketing. Additional access functions operating on the formula-tree return sub-structure information, such as bracketed sub-expressions, topographical parts (e.g. left/right side), top node for given substructures, etc.

For ill-formed expressions, a formula modification procedure starts with the set of formula trees and incrementally generates hypotheses by applying local, contextually justified modification rules in a best-first fashion, while testing whether the error is resolved. Each hypothesis is evaluated by a domain reasoner (for more details see [7]).

The final output of the formula analysis and correction procedure contains the following information passed as input to the response generation module: (1) the original expression with nodes marked for bracketing and type mismatches, (2) error source (see Table 2.), (3) the modifications applied to correct the expression, (4) the resulting corrected expression.

4. Generating Responses

When confronted with a formally flawed, but otherwise reasonable student contribution to the problem-solving task, the tutor has a number of options to react. In our Wizard-of-Oz studies, we have observed a variety of tutor reactions, categorized as follows (in the examples we always use English translations of the original German statements):

- *Refusal*: This is the simplest kind of reaction equivalent to minimal feedback.
- *Hinting*: This category constitutes an implicit rejection. It aims at giving the student some idea of where to look for spotting the error committed. Examples include "$=$ is not meaningful in this position" and "Do you really mean $a\cup(b\cap c))$?"
- *Meta-hinting*: This category is a kind of variant of the former one, where the content of the hint refers to some external material rather than properties of the student statement. An example is "Look into the course material about power set".
- *Assessment*: This category constitutes a very specific kind of hint, which amounts to a precise description of the consequence of the mistake made. This may be done either in terms of expressing the meaning of the student statement, or by describing the difference to the intended expression. An examples is "That would mean the union instead of the intersection.".
- *Clarification*: In this category, the student is asked to enhance his previous contribution or to make it more precise. Examples include "What do you mean with $PC \cup (A \cap B)$?", "The expression $a \cup b \cap c$ is ambiguous. Please clarify.".
- *Correction*: In this category, the tutor immediately reveals the error by providing the modified correct statement. An example is "No, this is not quite right. Apparently you mean $(b \cup a) \in P(a \cup b)$."

The proper generation of responses is done by instantiating partially embedded patterns consisting of frozen natural language text portions interleaved with specifications of references to further patterns. This process is directed by data produced by the error diagnosis and categorization module, and it is regulated by constraints imposed on the phrasing to be composed. Technically, this component is a strawman's version of the TG/2 surface generator [11]. This system is able to combine "shallow" and "deep" generation, that is, integrate surface-oriented string patterns with linguistically motivated unification-based language production. For our purposes, we do not need general feature path equations, but only access to attributes, and we do not make use of full unification since attributes of specifications produced by previous components are percolated down into appropriate places in natural language templates. There is, however, some nesting on subpatterns, but no recursion as in general linguistic structures, which enables the module to handle agreement and other limited forms of dependencies.

The patterns are of the form <LABEL>(<TEMPLATE>$^+$,<CONSTRAINT>*). The templates regulate the embedding in the response generation. In the general case, a template constitutes a reference to another pattern, with a label identifying this pattern, and a

list of parameters, which instantiate ordered constraints in the referred pattern. The simple cases, which end embedding of patterns, are either 1) an instantiation to a frozen natural language phrasing, 2) the insertion of a parameter specification, such as a formula, or 3) an access to a lexical item, including inflection information if required.

The lexical material available consists of three mini knowledge sources: 1) a proper lexicon for relevant mathematical objects, including inflection information, 2) a simple hierarchy, which expresses generalisation relations between mathematical objects (used, e.g., for references to parts of the domain description material), and 3) a "collocation" table, which maps properties of mathematical objects onto their contextually appropriate names, which may differ across mathematical objects (such as "sides" of an "equation", as opposed to "arguments" of ordinary functions).

In all top level patterns, the list of constraints includes, at least, the source of the error (see Table 2.) to be matched, and possibly tests for the availability of lexical material for a mathematical item to be referred to. Moreover, the constraints can also accommodate context-based criteria, such as attributes of preceding dialog contributions taken from the discourse memory, as well as attributes of tutoring strategies. Thus, the response category may not only depend on the category of the error made by the student, but also on tutoring strategies, reflecting the variation in actions taken by our tutors. Specifically, a change in the discourse context may lead to a variation in the reaction to a reoccurring error. A choice in instantiating these patterns lies in choosing the expression to be inserted. Specifically for hinting, a subexpression focusing on the error seems to be more appropriate than the full expression given by the student. Mimicking the choices of the tutors, only the erroneous operator, if unique in the student statement, is picked, or the subformula in which the operator or one or several of its arguments is erroneous. These choices are expressed by referring to the appropriate formula-tree access functions (see the previous section) in the patterns.

With the given specifications, we can generate most of the tutor responses to student errors in formulas as observed in our corpora. Exceptions concern advanced referring expressions, which include some vagueness in the description, such as "the *second half* of the formula", and exploitation of conversational implicature, such as "the *second term*", which implicitly means the second term on top level of the referred formula, which several "second" terms embedded in this formula.

5. Evaluation

In our corpora, simple refusals were seen quite frequently, in many, but by far not all cases, forced by the prescribed tutorial strategy. Clarification responses were used in cases of ambiguity or when problem-solving steps were too coarse-grained. Hinting as characterized above was typically used to address syntax or type errors, sometimes also for semantic errors. All other response types were seen much less frequently since they require some kind of specific circumstances. Meta-hinting was applied in connection with erroneous applications of axioms (their instantiation) leading to semantic errors. Building an assessment of the student statement also occurred infrequently since its suitable usage requires that the error follows a straightforward realization so that a simple enough verbalization can be built. Confirmation, finally, was observed in cases where the tutor assumed the student to know the right statement and failure to produce it was only due to some lapse, such as a typing error.

We performed a preliminary qualitative evaluation of the feedback generation algorithm on a sample of flawed formulas from Corpus-I: we constructed a questionnaire of single-turn dialog responses to 15 flawed expressions (with different error sources). The responses were generated according to our feedback strategies based on the output of the mathematical expression analysis module. As control condition we used the *Refusal* response that included only the information that the expression was incorrect.

We asked 4 mathematics tutors to fill out the questionnaire. We asked about pedagogical appropriateness of the two responses (*minimal* vs. our algorithm). The participants were explicitly instructed to indicate which response they considered more appropriate. Furthermore, we asked the participants to add their own response proposal, in case they could think of a more informative reaction, and to justify their opinion.

Overall, tutors agreed with our response proposals favoring the response generated by our algorithm over *minimal feedback*. Three tutors consistently either selected our feedback or proposed their own. One tutor chose the *minimal feedback* response over ours in 7 out of 15 cases, however, explicitly commented on the fact that a *minimal feedback* response would be preferred as the first reaction with a more explicit feedback provided if the student is not capable of locating and fixing the error themselves.

6. Related Work

Tutorial systems addressing faulty statements in a dialog can essentially be partitioned in three categories: approaches assessing complete solution proposals ([12], [13] and [14]), approaches dealing with answers in some restricted dialog context ([15] and [16]), and approaches that allow students to express their next problem-solving step in a rather free manner according to the given discourse situation ([17] and [18]). Our model falls into the third category. As opposed to approaches addressing complete solutions, we aim at dealing with formal inaccuracies in view of the problem-solving context, rather than exposing violated constraints [12]. or pointing to argumentative inconsistencies [14]. A crucial distinction to the competing approaches lies in its architecture which comprises distinct components for performing error analysis and correction, and for building responses. Specific contributions of our method lie in the flexible analysis component which can handle chaining of formal expressions, formulas interleaved with portions of natural language, and has access to a generic reasoning component.

7. Conclusion

In this paper, we have presented a model for generating responses to formally buggy, but otherwise reasonable problem-solving statements containing mathematical formulas. The elaboration of messages that can be generated is based on evidence from observing human tutors in Wizard-of-Oz studies on tutoring mathematical theorem proving. Our method comprises error detection and correction attempts aiming at conceptual error categorization which, in turn, gears proper message generation. Moreover, we can accommodate several tutoring strategies, so that only components of the model need to be changed to adapt it. In the future, we intend to explore the flexibility of our model. Apart from adapting the error-related component to other subdomains of mathematics, we intend to increase its natural language capabilities – specifically, more elaborate referring expressions to formulas and their components, according to [19].

References

[1] C. Benzmüller, A. Fiedler, M. Gabsdil, H. Horacek, I. Kruijff-Korbayová, M. Pinkal, J. Siekmann, D. Tsovaltzi, B. Q. Vo, and M. Wolska. Tutorial dialogs on mathematical proofs. In *Proc. of IJCAI Workshop on KR and AR for E-Learning Systems*, pp. 12–22, 2003.

[2] M. Wolska, B.Q. Vo, D. Tsovaltzi, I. Kruijff-Korbayová, E. Karagjosova, H. Horacek, M. Gabsdil, A. Fiedler, and C. Benzmüller. An annotated corpus of tutorial dialogs on mathematical theorem proving. In *Proc. of LREC-04* , pp. 1007–1010, Lisbon, Potugal, 2004.

[3] C. Benzmüller, H. Horacek, H. Lesourd, I. Kruijff-Korbayová, M. Schiller, and M. Wolska. A corpus of tutorial dialogs on theorem proving; the influence of the presentation of the study-material. In *Proc. of LREC-06* , pp. 1766–1769, Genoa, Italy, 2006.

[4] M. Wolska and I. Kruijff-Korbayová. Factors influencing input styles in tutoring systems: the case of the study-material presentation format. In *Proc. of the ECAI-06 Workshop on Language-enabled Educational Technology*, Riva del Garda, Italy, 2006.

[5] H. Horacek and M. Wolska. Interpretation of mixed language input in a mathematics tutoring system. In *Proc. of the AIED-05 Workshop on Mixed Language Explanations in Learning Environments*, Amsterdam, the Netherlands, 2005.

[6] C. Benzmüller, H. Horacek, I. Kruijff-Korbayová, H. Lesourd, M. Schiller, and M. Wolska. DiaWozII – A Tool for Wizard-of-Oz Experiments in Mathematics. In *Proc. of the 29th Annual German Conference on Artificial Intelligence (KI-06)*, Bremen, Germany, To Appear.

[7] H. Horacek and M. Wolska. Handling errors in mathematical formulas. In Mitsuru Ikeda and Kevin D. Ashley and Tak-Wai Chan, eds., *LNCS*, vol. 4053, pp. 339–348, 2006. *Proc. of the 8th International Conference on Intelligent Tutoring Systems (ITS-06)*, Jhongli, Taiwan.

[8] C. E. Brown. Verifying and invalidating textbook proofs using scunak. In *Mathematical Knowledge Management (MKM-06)*, pp. 110–123, Wokingham, England, 2006.

[9] J. Siekmann et al. Proof development with ΩMEGA. In Andrei Voronkov, editor, *Automated Deduction — CADE-18*, LNAI 2392, pp. 144–149. Springer Verlag, 2002.

[10] M. Wolska and I. Kruijff-Korbayová. Analysis of Mixed Natural and Symbolic Language Input in Mathematical Dialogs. In *Proc. of the 42nd Meeting of the Association for Computational Linguistics (ACL-04)*, pp. 25–32, Barcelona, Spain, 2004.

[11] S. Busemann and H. Horacek. A flexible shallow approach to text generation. In Eduard Hovy, editor, *Proc. of the Ninth International Workshop on Natural Language Generation*, pp. 238–247. Association for Computational Linguistics, Niagara-on-the-Lake, Canada, 1998.

[12] P. Suraweera, and A. Mitrovic. KERMIT: A constraint-based tutor for database modeling. In *Proc. of the 6th International Conference on Intelligent Tutoring Systems (ITS-02)*, pp. 671–680, Biarritz, France, 2002.

[13] K. VanLehn et al. The architecture of Why2-Atlas: A coach for qualitative physics essay writing. In *Proc. of the 6th ITS Conference (ITS-02)*, pp. 158–167, Biarritz, France, 2002.

[14] J. Toth, D. Suthers, and A. Weiner. Providing expert advice in the domain of collaborative scientific inquiry. In *Proc. of the 8th World Conference on Artificial Intelligence in Education (AIED-97)*, pp. 302–306, Kobe, Japan, 1997.

[15] Autotutor, 2006. `http://www.autotutor.org/`.

[16] N. T. Heffernan and K. R. Koedinger. An intelligent tutoring system incorporating a model of an experienced human tutor. In *Proc. of the 6th International Conference on Intelligent Tutoring Systems (ITS-02)*, pp. 596–608, Biarritz, France, 2002.

[17] Circsim-tutor, 2002. `http://www.csam.iit.edu/~circsim/`.

[18] K. VanLehn, C. Lynch, K. Schulze, J. Shapiro, and R. Shelby. The andes tutoring system: Five years of evaluations. In *Proc. of the 12th International Conference on Artificial Intelligence in Education (AIED-05)*, pp. 678–685. IOS Press, 2005.

[19] H. Horacek. Generating references to parts of recursively structured objects. In *Proc. of the Fourth International Natural Language Generation Conference (INLG-06)*, pp. 47–54, Sydney, Australia, 2006. Association for Computational Linguistics.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Semantic Web Technology to Support Learning about the Semantic Web

Martin DZBOR and Enrico MOTTA
Knowledge Media Institute, The Open University, Milton Keynes, UK

Abstract. In this paper we describe ASPL, Advanced Semantic Platform for Learning, designed using the Magpie framework with an aim to support students learning about the Semantic Web research area. We describe the evolution of ASPL and illustrate how we used the results from a formal evaluation of the initial system to re-design the user functionalities. The second version of ASPL semantically interprets the results provided by a non-semantic web mining tool and uses them to support various forms of semantics-assisted exploration, based on pedagogical strategies such as performing *lateral steps* and *query filtering*.

Keywords: Semantic Web framework, Magpie, e-learning support, annotation

Introduction

Education, like other disciplines, takes advantage of the Web to provide learning resources speedily and easily, and to tailor them to the specific needs of a learner. However, education has always relied on a strong *interpretative component*. In addition to recalling knowledge from knowledge bases, searching document repositories or retrieving from information warehouses, education also relies on analysis and synthesis – both on the level of individual learners and at group level. Interpretation, in general, comprises the ability to link otherwise independent information, to make statements about it, and to make inferences from the available knowledge. Also, education is an interactive activity that centers on the learners and expects their active participation.

The size of databases and other repositories of resources that are suitable for learning is no longer the greatest obstacle in harnessing the Web in educational practice. Already in 1940s Bush [3] pointed out there was more information published than it was possible for humans to process. This processing bottleneck has not been overcome yet; in learning it often takes the form of relating one chunk of knowledge to another and applying it in new situations. Similarly, Bloom [1] argued that information processing goes beyond its recall and that advanced cognitive processes, such as synthesis and judgment lead to longer-lasting knowledge. These processes share one feature: they are *relational*; i.e., they comprise associations between separate pieces of information.

Although humans rely on associations to make sense of information, the challenge of supporting associative thinking is far from resolved. For instance, in [9] the authors point out that as (i) it is hard to formally capture all subtleties of a learning task in a tutoring system, and (ii) learner modeling is only approximate, tutoring systems tend to be over-constrained closed worlds. To address the constraints, it is more valuable for the learner to see *what can be done* with a given knowledge, rather than merely following a prescribed workflow for a learning task.

In this paper, we present an approach to designing a semantically enriched application, the *Advanced Semantic Platform for Learning* (ASPL), to facilitate exploratory learning on the Web. We summarize results of evaluating a data retrieval centred ASPL-v1, and justify design amendments for ASPL-v2. We conclude by situating this work in the wider context of supporting learning and exploration on the Web, and we highlight a number of principles about the effective use of Semantic Web technologies in this context.

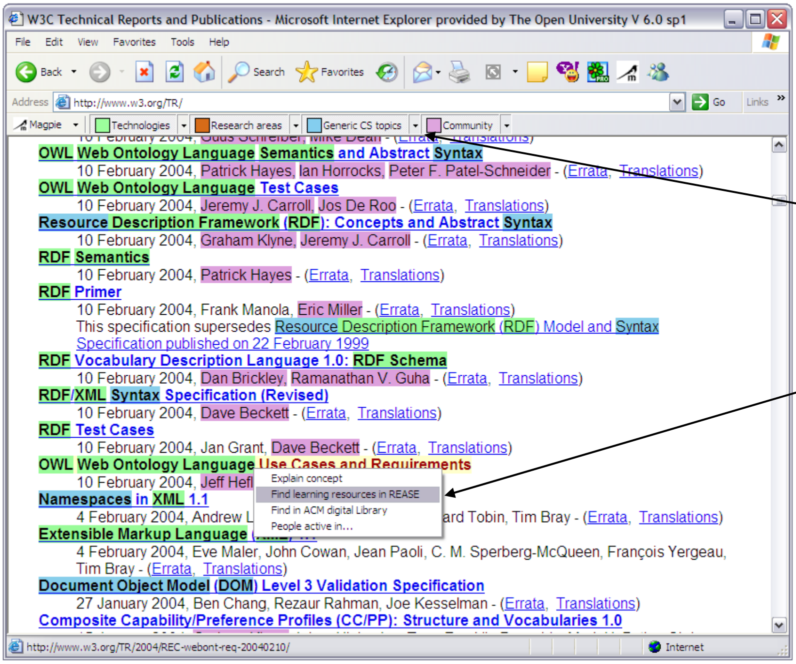


Fig. 1. A screenshot showing a Magpie-enhanced web browser on a visited web page, which is annotated using the lexicon semi-automatically acquired for the Semantic Web domain

1. Evaluating a Semantic Web application for learning support

Fig. 1 shows an initial version of our experimental system (ASPL-v1) with an ontology about the Semantic Web research and community and applied to the web page of W3C reports library (http://w3.org/TR). ASPL is based upon our *Magpie* framework [7], which offers a generic solution to partially address a known knowledge acquisition bottleneck [11]: namely, how to apply ontologies to the task enrichment (in our case, to the selection and navigation through learning resources), and how to accomplish this in the absence of extensive semantic annotations of such resources.

In terms of learning, ASPL intends to offer learners the access not only to atomic resources (e.g. publications) but also to relational associations. Our aim is to show that semantically associated entities enable the learner to draw more abstract analytic and/or synthetic conclusions, which are, in turn, beneficial to support an open-ended task of analyzing and reviewing the state-of-the-art in a given research domain.

ASPL as a Magpie-based application takes a user-selected lexicon, and uses it to annotate web pages, as the user browses the Web. The key aspects of Magpie user interface

are shown by markers ❶ and ❷ in Fig. 1; the semantic layers are shown as colour highlights over standard, non-semantic resources. Technical details of Magpie appeared elsewhere [7] and are not subject of this paper.

The ASPL-v1 experimental system was bootstrapped with ontologies describing the Semantic Web research and community. The 'Community' and 'Research areas' categories were *automatically* populated using a tool for mining and capturing entities from text corpora [15]. With this prototype we carried out a user study around the task of preparing a literature review on a given subject. To prevent confounding the study with our views on how participants should carry out the task, we constrained the services to information retrieval: publications were shown with no further guidance on what to do next provided.

Table 1. Overall and per-location results of ASPL-v1 evaluation

	Mean quality		*Overall mean score*		*Significance*
	A	*B*	*A*	*B*	*t-test (overall)*
Overall	**10.85**	**9.79**	**19.11**	**18.19**	**0.2851003**
Site 1	13.67	10.29	24.11	20.68	0.06526177
Site 2	12.00	8.00	20.25	13.82	0.06365032
Site 3	8.57	12.50	15.57	22.26	0.00469364
Site 4	8.86	8.00	15.57	14.71	0.39071656

1.1. Formative evaluation and requirements gathering

Our evaluation took place in November 2005 at four universities in Europe, referred below as Site 1 to 4. We aimed (i) to see how semantic annotations and services are used by students during an information gathering task, and (ii) to identify ways of supporting more complex parts of the task of preparing the literature review using the Semantic Web. Group A comprised participants using Google; thus this was our control group. Group B involved participants using ASPL-v1.

The participants were asked to compile ten references, which in their view addressed the task they were given. Their interactions with the ASPL-v1 were recorded, and responses were marked by a domain experts based on their subjectively perceived suitability for a literature review. A participant's score was the sum of all the marks, and is shown in Table 1. The 'Mean quality' reflects only the marks from the expert; whereas the 'Overall mean score' takes into account time limit bonuses/penalties.

The variance in performance was not statistically significant for the whole population (at p=5%) and only 'Site 3' showed a performance rise for the ASPL users. As we suspected, skilled users of Web search engines did not find much value in Magpie-annotations solely for retrieving but not discriminating the results – similarly to generic search engines. The outlier 'Site 3', however, comprised students with a prior tuition in writing literature reviews. They relied on semantic annotations more frequently and tried to interpret and explore the retrievals through Magpie services. The annotations helped them filter and more cleverly navigate to the key references among retrievals.

If we qualitatively generalize this – our outliers went beyond mere information retrieval, whether semantic or not. They showed expectations for more *investigative navigation*, which was what we aimed to capture and embed in the ASPL. For example, these outliers expected to obtain some guidance on what to do with partial solutions, assistance with query changes and re-formulations, etc. A flat list of semantically

indiscriminate records provided in the initial demonstrator of ASPL was clearly insufficient in this respect.

1.2. Conceptual analysis of the learning task

In our view, people who managed to replicate some aspects of the investigative or *exploratory navigation* through the retrieved resources gained more from Semantic Web annotations. The challenge for the ASPL re-design was thus to facilitate more of the guided exploration in otherwise large and ill-defined problem space. If we look at the task our participants tackled, its outcome is essentially an argument comprising publications and rationale that can be conceptualized into a literature review 'model'.

As our user study showed, there may be many configurations of such a task model. We call these configurations *sensemaking paths*. A sensemaking path, along which learning tasks can be organized, is one way to elaborate and to formalize exploratory navigation. Here, exploration corresponds to the capability to refine an initial problem; it helps reduce the number of retrieved papers if (say) teams of co-authors or broader communities of practices are explored as alternatives to the original query.

The alternative paths differ in how the review task is interpreted – is it more about listing prominent approaches or is it about comparing alternative approaches? Is it author- or topic-driven? Both interpretations are valid and accomplish the same aim. In our study, people tended to start with a literal (i.e. topic-driven) interpretation of the task. If the search output was too coarse-grained and semantically generic, an alternative strategy could be; e.g.: (i) filtering key authors and following that trail to specific works, or (ii) looking at topics conceptually close to the ambiguous original query to alter the initial problem. The choice between alternative sensemaking paths uses pedagogic knowledge and guidance, which, in turn, enables us to support the learner in performing a 'lateral' step from one sensemaking strategy to another; e.g. from a topic-driven to the author-driven one. It frames an open task, and it is orthogonal to the task decomposition that is common in many intelligent tutoring systems (ITS).

Benefits of Semantic Web to a user's interaction with learning resources are in an improved interpretation of the navigational choices in a multi-dimensional space. At any point, there are alternative directions to take; each step triggers a specific outcome, has specific strengths and weaknesses. Some are about *deepening knowledge in one direction* (this is conceptually close to faceted views [14]) and other steps are *'lateral'*; i.e. moving from one view of the domain to another (similar to horizontal navigation [2]). Unlike specialized faceted or navigational tools, Semantic Web technology may address both alternatives and offer a more robust and flexible support for the learner.

2. How to realize different sensemaking strategies

The notion of exploratory navigation is not novel [3, 4, 9]. Our work brings to the state of the art an *open, extensible framework* and the capability to achieve *exploration by combining the specialized services*. Information retrieval services are rarely conceived as exploratory; their aim is to access material needed for a given task. The ASPL framework offers several entry points enabling the exploration by linking such simple services *and* interpreting their outcomes semantically. For example, entities such as authors or research areas in ASPL provide means to get started with the literature

review. The role of ASPL is to enrich these ‘gateways’ by offering services and realize the filtering, deepening and lateral dimensions, as mentioned in section 1.2.

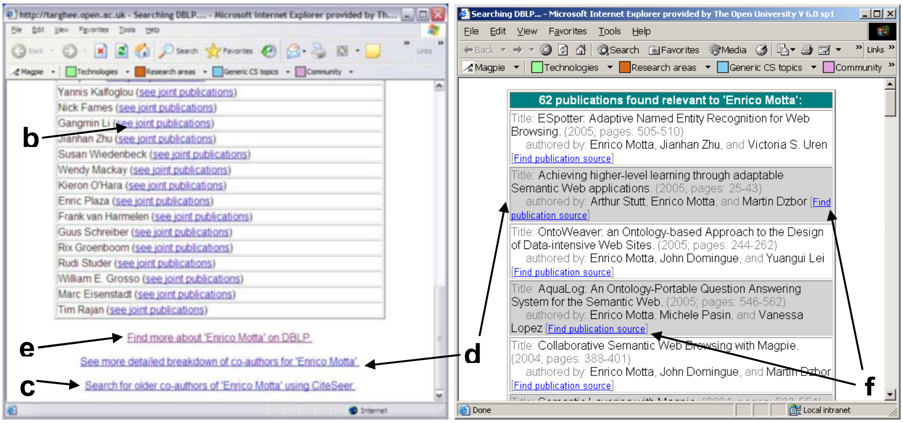


Fig. 2. Screenshots with different forms of ‘path narrowing’ as illustrations of sensemaking strategies

2.1. Strategy #1: filtering acquired knowledge

The filtering strategy was implemented to refine the response to the query satisfying a given keyword. On the cognitive level, what these refinements aimed at was the ‘spotlight navigation’ – continuously narrowing a path towards something that can be further analyzed. This path narrowing strategy is not hierarchic or taxonomic; it lacks the prescriptive nature such structures. For example, if the learner requests a list of co-authors publishing with Enrico Motta (see Fig. 2), the ASPL-v2 sensemaking support now offers to refine the initial list of co-authors (not shown) in several directions:

- filter publications to those co-authored with a specific person (Fig. 2b)
- look at the co-author community over time; e.g. older vs. new clusters (Fig. 2c)
- view co-authorship in terms of publication clusters vs. individuals (Fig. 2d)
- explore the provenance and hence credibility of the co-author list (Fig. 2e)
- formulate query for finding the source of a particular publication (see Fig. 2f)

As shown in Fig. 2 these alternative ways of continuing the exploration of the retrieved resources use different semantic relationships. For instance, link in Fig. 2b uses a relation of co-authorship (applicable to a person); link in Fig. 2c deploys the property of publication year (applicable to papers). None of these is sophisticated on its own, but such independent chunks can be easily mined in documents or in databases. The added value of Semantic Web is in inferring meaningful associations among such chunks.

2.2. Strategy #2: lateral navigation

On the other hand, the strategy of lateral navigation is an opposite to the ‘spotlight browsing’ – a serendipitous and divergent exploration of different aspects of a particular collection. The purpose of modelling this strategy was not to narrow the exploratory path, but on the contrary, to open it up in a loosely related but different direction. This learning strategy responds to the question of *how else* a given task can be achieved.

For example, a learner may start with retrieving publications using “hypertext” as a keyword, and gets thousands of references containing that keyword. In ASPL-v1, the collection used in this manner was ACM Portal (http://portal.acm.org), which offered 40,000+ hits in response queries of this type. A remedy to receiving too large a number of results is e.g. one of the alternative approaches to the problem:

1. try expanding or changing “hypertext” e.g. with *related topics*, or
2. try shifting to *leading authors* in “hypertext” and possibly search for the publications of those towards the top of the list, or their co-authoring communities

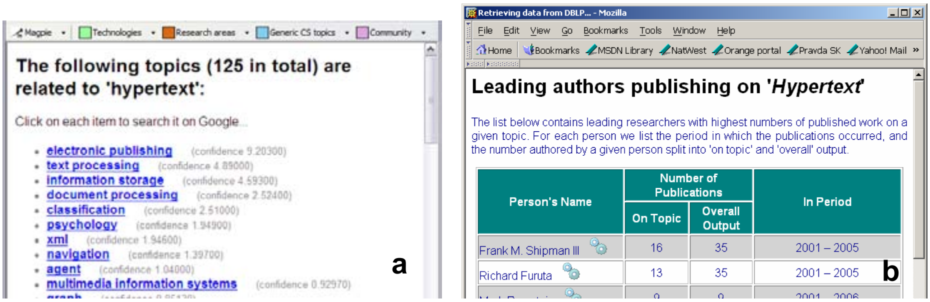


Fig. 3. Lateral navigation from a topic-based query: (a) to related sub-topics, and (b) to experts

Alternative reformulations of the original problem are shown in Fig. 3: on the left is an automatically mined list of themes related to “hypertext”, while on the right is an orthogonal list of authors active in “hypertext”. Fig. 3a can be interpreted: *Explore main topic “hypertext” in the context of sub-topic (e.g.) “navigation”*. Moreover, one can also carry out more analytic or synthetic tasks, such as for example: *Compare the results of “navigation” vs. “text processing” contexts of the main theme “hypertext”*. Similarly, by switching from the topics to authors, one actually reformulates the original retrieval problem using semantically close entities. This, in turn, leads to different results – a shortcut to these queries is expressed by the ‘cogs’ icon metaphor in Fig. 3b.

2.3. Benefits of an automated discovery of associations

ITS [10] and other learner-support tools are active applications – leading the learner through a content. Yet they largely operate on a closed set of resources and rely on manually defined abstract relationships between concepts and resources [9]. The most popular link of this kind is ‘requires’ – as in “*Study of ontology-based annotation requires knowledge of RDF*.” Applying the ‘requires’ link transitively, one may theoretically compute possible learning paths through the resources and ensure the user follows a prescribed learning task. However, manual annotation assumes one path fits all users and is not scalable. This bottleneck reduces the feasibility of ITS.

Rather than tying the learner into a specific learning task, we see learning tasks as an optional element in a semantic system guiding the learner. Many learning tasks can be achieved by following distinct *paths through the space of (learning) resources*. It is nearly impossible to formalize any one of these paths as an ideal execution of the learning task. Instead, different paths can be triggered by associations that are useful at a particular moment. The relationship between tasks and paths is many to many – one task can be achieved by following several paths, and vice versa.

3. Learning and exploration: Related research

The notion of exploratory user interaction with learning resources has been around for some time. In the pedagogical domain, for example Laurillard [12] sees learning as a conversation of the learner with the problem and available resources – a conversation that facilitates exploration of relevant knowledge spaces and creates a rich set of different kinds of associations between the nodes in the explored knowledge space.

Similar insights have been made by cognitive scientists studying skill acquisition in engineering design. For example, Schön [13] talks about reflective conversations of a practitioner with a design situation. These allow multiple ways of interpreting the situation and rely on associative frames. The theory of problem framing [6, 13] was developed on a rather abstract, cognitive level; nevertheless, its core is in recognizing and selecting associations to make sense of a situation. This, in turn, can be considered as the basis for a more operational, technical approach.

The Web community attempted to address the issue of dynamic associations in standard hypermedia. E.g. COHSE [4] has re-introduced the serendipity into navigating using open hyperlinks. Its hyperlinks were to some extent independent of the actual web pages and could be inserted as and when a need arose. This approach is known as open hypermedia. In educational hypermedia, use of horizontal (non-hierarchical) navigation in digital textbooks was analyzed in [2]. A *lack of support for the horizontal links* was noticed, as this mode of navigation is more resource-intensive than classification into a vertical taxonomy. Vertical content-based links represent the plan; the horizontal associative links are more serendipitous, interpreted, and may be subjective to a learner.

Nonetheless, one obstacle has always been the problem of specifying meaning. In the standard Web, this is normally implicit inside the mind of the resource provider. Semantic Web technologies, such as RDF or OWL[1], take a complementary stance by assuming that each relation or association can be committed to a formal semantic interpretation. Once an explicit semantic commitment (or annotation) is made, the relation acquires a specific meaning, which in turn enables distinguishing between (say) a person being active in an area and one area being similar to another. In short, the Semantic Web extends the notion of associative navigation by the fact that associations are not only established, but more importantly, interpreted.

4. Conclusions

Our research perceives a learner's interaction with resources on the Semantic Web as more than annotation, retrieval and subsequent browsing of semantic metadata. In order to apply semantic knowledge the re-designed ASPL-v2 used an exploratory approach to interact with distributed learning resources. Specifically we tested two distinct modes of exploratory learning: (i) convergent, 'spotlight' browsing of semantically enriched resources [5], and (ii) divergent, 'serendipitous' browsing into an open web space [8].

As a source of information, knowledge and guidance, more and more used to support learning both formally and informally, the Web needs tools with the capacity to create semantic associations and intelligently use such associations e.g. to filter more

---

[1] See W3C recommendations http://www.w3.org/TR/rdf-schema and http://www.w3.org/TR/owl-ref.

familiar user interaction processes, such as search or data retrieval. The main advantage of ASPL – annotation tied in with relationship composition is in reducing information overload and in increasing the smartness of the systems supporting learners.

Applying Semantic Web to construct multiple exploratory paths and attending to different aspects of the exploration, rather than to the individual nodes of the semantically enriched space, has several side effects. For instance, from the user experience viewpoint, the application becomes more flexible. A semantically enriched application does not confine its user to one specific activity or role. Another side effect is the dynamics of the semantic application. Ontology-driven solutions are often brittle; often based on closed worlds that enable reasoning solely about the known concepts. Linking the association discovery to the presentation overcomes this brittleness, and also avoids the knowledge acquisition bottleneck.

Acknowledgements

This work is supported by the KnowledgeWeb and NeOn projects funded by the IST Framework VI of the European Commission's grants no. FP6-507482 and FP6-027595.

References

[1] Bloom, B.S., *A Taxonomy of Educational Objectives Handbook 1: Cognitive Domain*. 2nd ed. 1965, New York, US.

[2] Brusilovsky, P. and Rizzo, R., *Map-Based Horizontal Navigation in Educational Hypertext.* Journal of Digital Information, 2002. **3**(1): p. 156.

[3] Bush, V., *As we may think.* The Atlantic Monthly, 1945. **176**(1 (July)): p. 101-108.

[4] Carr, L., Bechhofer, S., Goble, C*., et al. Conceptual Linking: Ontology-based Open Hypermedia*. In *10th Intl. WWW Conf.* 2001. Hong-Kong.

[5] Collins, T., Mulholland, P., and Zdrahal, Z. *Semantic Browsing of Digital Collections*. In *Proc. of the 4th Intl. Semantic Web Conf.* 2005. Ireland. p. 127-141.

[6] Dzbor, M. and Zdrahal, Z. *Design as interactions between problem framing and problem solving*. In *15th European Conference on AI (ECAI)*. 2002. Lyon, France. p. 210-214.

[7] Dzbor, M., Motta, E., and Domingue, J. *Opening Up Magpie via Semantic Services*. In *Proc. of the 3rd Intl. Semantic Web Conf.* 2004. Japan. p. 635-649.

[8] Dzbor, M., Motta, E., and Stutt, A., *Achieving higher-level learning through adaptable Semantic Web aplications.* Int. J. of Knowledge and Learning, 2005. **1**(1/2): p. 25-43.

[9] Eisenstadt, M., Price, B.A., and Domingue, J., *Software Visualization As A Pedagogical Tool.* Instructional Science, 1983. **21**: p. 335-365.

[10] Frasson, C., Gauthier, G., and McCalla, G.I. *Intelligent Tutoring Systems*. In *Intl. Conf. on Intelligent Tutoring Systems (ITS)*. 1992: Springer-Verlag, Berlin.

[11] Hayes-Roth, F., Waterman, D.A., and Lenat, D.B., eds. *Building Expert Systems*. 1983, Addison-Wesley: Massachusetts, US.

[12] Laurillard, D., *Rethinking University Teaching: A Conversational Framework for the Effective Use of Learning Technologies*. 2nd ed. 2002, London, UK: RoutledgeFarmer.

[13] Schön, D.A., *Reflective Practitioner - How professionals think in action*. 1983, USA: Basic Books, Inc.

[14] Schraefel, M.C., Smith, D.A., Owens, A*., et al. The evolving mSpace platform: leveraging the Semantic Web on the Trail of the Memex*. In *Proc. of Intl. Conf. on Hypertext*. 2005. Austria.

[15] Zhu, J., Goncalves, A.L., Uren, V.S*., et al. Mining Web Data for Competency Management*. In *Proc. of the IEEE/WIC/ACM Conf. on Web Intelligence*. 2005. France.

Exploring Neural Trajectories of Scientific Problem Solving Skill Acquisition

Ronald H. STEVENS[1], Trysha GALLOWAY[1], Chris BERKA[2]
[1] UCLA IMMEX Project, 5601 W. Slauson Ave. #255, Culver City, CA 90230
immex_ron@hotmail.com,
tryshag@gmail.com
[2] Advanced Brain Monitoring, Inc, Carlsbad, CA 90045
chris@b-alert.com

Abstract. We have modeled changes in electroencephalography (EEG) - derived measures of cognitive workload, engagement, and distraction as individuals developed and refined their problem solving skills in science. Subjects performing a series of problem solving simulations showed decreases in the times needed to solve the problems; however, metrics of high cognitive workload and high engagement remained the same. When these indices were measured within the navigation, decision, and display events in the simulations, significant differences in workload and engagement were often observed. In addition, differences in these event categories were also often observed across a series of the tasks, and were variable across individuals. These preliminary studies suggest that the development of EEG-derived models of the dynamic changes in cognitive indices of workload, distraction and engagement may be an important tool for understanding the development of problem solving skills in secondary school students.

Introduction

Skill development occurs in stages that are characterized by changes in the time and mental effort required to exercise the skill (Anderson, 1982, 1995, Schneider and Shiffrin, 1977). Given the complexities of skill acquisition it is not surprising that a variety of approaches have been used to model the process. For instance, some researchers have used machine learning tools to refine models of skill acquisition and learning behaviors in science and mathematics. Such systems rely on learner models that continually provide updated estimates of students' knowledge and misconceptions based on actions such as choosing an incorrect answer or requesting a multimedia hint. Although such learner models are capable of forecasting student difficulties (Stevens, Johnson, & Soller, 2005), or identifying when students may require an educational intervention, they still rely on relatively impoverished input due to the limited range of learner actions that can be detected by the tutoring system (e.g., menu choices, mouse clicks).

There is a large and growing literature on the EEG correlates of attention, memory, and perception (Fabiani, 2001, Smith, 1999, Berka, 2004, Berka 2006). However, EEG researchers have generally elected to employ study protocols that utilize training-to-criterion to minimize variability across subjects and ensure stable EEG parameters could be characterized. In most studies, the EEG data is not even acquired during the

training process, leaving an untapped and potentially rich data source relating to skill acquisition.

While advanced EEG monitoring is becoming more common in high workload / high stress professions (such as tactical command, air traffic controllers), the ideas have not been comprehensively applied to real-world educational settings due to multiple challenges. Some of these challenges are: 1) the acquisition of problem solving skills is a gradual process and not all novices solve problems in the same way, nor do they follow the same path at the same pace as they develop domain understanding; 2) given the diversity of the student population it is difficult to assess what their relative levels of competence are when performing a task making it difficult to accurately relate EEG measures to other measures of skill and 3) the strategic variability makes analyzing the patterns of students' problem solving record too complicated, costly, and time consuming to be performed routinely by instructors; nevertheless, there are many aspects of science education that could benefit from deriving data from advanced monitoring devices and combining them with real-time computational models of the tasks and associated outcomes conditions.

This manuscript describes a beginning synthesis of 1) a probabilistic modeling approach where detailed neural network modeling of problem solving at the population level provides estimates of current and future competence, and, 2) a neurophysiologic approach to skill acquisition where real-time measures of attention, engagement and cognitive work load dynamically contribute estimates of allocation of attention resources and working memory demands as skills are acquired and refined.

1. Methods

1.1. The IMMEX™ Problem Solving Environment

The software system used for these studies is termed IMMEX™ which is based on an extensive literature of how students select and use strategies during scientific problem solving (VanLehn, 1996, Haider & Frensch, 1996).

To illustrate the system, a sample biology task called *Phyto Phiasco* provides evidence of a student's ability to identify why the local potato plants are dying.

The problem begins with a multimedia presentation explaining the scenario and the student's challenge is to identify the cause. The problem space contains 5 Main Menu items which are used for navigating the problem space, and 38 Sub Menu items describing local weather conditions, soil nutrients, plant appearance, etc. These are decision points, as when the student selects them, s/he confirms that the test was requested and is then presented the data. When students feel they have gathered the information needed to identify the cause they attempt to solve the problem.

The IMMEX database serializes timestamps of how students use these resources. As students solve IMMEX cases, the menu items selected are then used to train competitive, self-organizing ANN (Stevens & Najafi, 1993, Stevens et al, 1996). We use the outputs of this classification to define strategic snapshots of each student's performance. Students often begin by selecting many test items, and consistent with models of skill acquisition (Ericsson, 2004), refine their strategies with time and select fewer tests, eventually stabilizing with an approach that will be used on subsequent problems. As expected, with practice solve rates increase and time on task decreases. The rate of stabilization, and the strategies stabilized with are influenced by gender

(Stevens & Soller, 2005), experience (Stevens et al, 2004), and individual or group collaboration.

IMMEX problem solving therefore represents a task where it is possible to construct probabilistic models of many different aspects of problem solving skill acquisition across problem solving domains. The constraints of working memory are likely to be particularly relevant during such skill acquisition where working memory capacity can frequently be exceeded. The possibility of combining these models with EEG workload metrics raises questions regarding the cognitive demands and the balances of different working memory capacities needed as students gain experience and begin to stabilize their strategies?

1.2. The B-Alert®system

A recently developed wireless EEG sensor headset has combined a battery-powered hardware and sensor placement system to provide a lightweight, easy-to-apply method to acquire and analyze six channels of high-quality EEG. Standardized sensor placements include locations over frontal, central, parietal and occipital regions (sensor sites: F3-F4, C3-C4, Cz-PO, F3-Cz, Fz-C3, Fz-PO). Data are sampled at 256 samples/second with a bandpass from 0.5 Hz and 65Hz (at 3dB attenuation). Quantification of the EEG in real-time, referred to as the B-Alert® system, is achieved using signal analysis techniques to identify and decontaminate fast and slow eye blinks, and identify and reject data points contaminated with excessive muscle activity, amplifier saturation, and/or excursions due to movement artifacts. Wavelet analyses are applied to detect excessive muscle activity (EMG) and to identify and decontaminate eye blinks.

1.3. Subjects and Study

Subjects (n=12) first performed a single 30-minute baseline EEG test session to adjust the software to accommodate individual differences in the EEG (Berka, 2004). They then performed multiple IMMEX problem sets targeted for 8th-10th grade students. These include *Phyto Phiasco*, the biology problem described above and a mathematics problem called *Paul's Pepperoni Pizza Palace*. Subjects generally performed at least 3 cases of each problem set allowing the tracking of strategies and cognitive changes across cases as students gained experience. Then we aligned the EEG output metrics on a second-by-second basis with the problem solving actions to explore the within-task EEG metric changes. For this alignment, we used software (Morea, Techsmith, Inc.) that captures output from the screen, mouse click and keyboard events as well as video and audio output from the users (Figure 1).

The output of the B-Alert software includes EEG metrics (ranging from 0.1-1.0) for distraction (HDT), engagement (HE), and workload (HWL) calculated for each 1-second epoch using quadratic and linear discriminant function analyses of model-selected EEG variables derived from power spectral analysis of the 1-Hz bins from 1-40Hz.

These metrics have proven utility in tracking both phasic and tonic changes in cognitive states, and in predicting errors that result from either fatigue or overload (Berka 2005). The cognitive indices are expressed as histograms for each 1-second epoch of the problem solving session and show the probability of HWL, HE, or HDT. By integrating the time stamps of data requests with those of the B-Alert system, the

navigation, decision, and display-related events can be overlaid onto the cognitive indices.

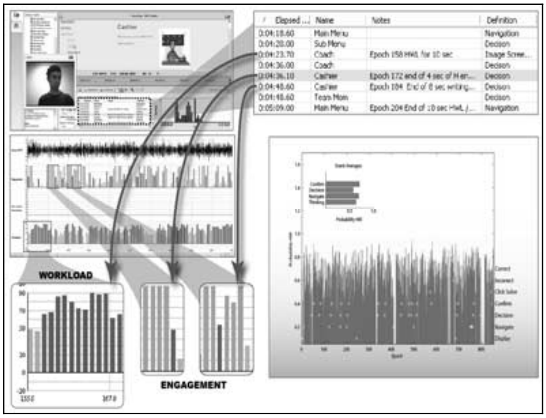


Figure 1. Relating EEG Workload and Engagement Indexes with Problem Solving Events. The upper left figure is a screen shot of a user (not described in the text) engaged in IMMEX problem solving while keyboard and mouse events are simultaneously recorded; below shows the real-time output of the B-Alert cognitive indexes. Samples of the workload and engagement data streams have been linked with specific events in the log.. In the lower right corner, the timestamps of IMMEX data requests and displays are integrated with the EEG workload indices and then plotted against the one-second epochs of the task. The upper left histograms average the workload indices for each of the IMMEX events including the one second prior to and after the event.

2. Distributions of Engagement, Distraction and Workload During IMMEX Problem Solving

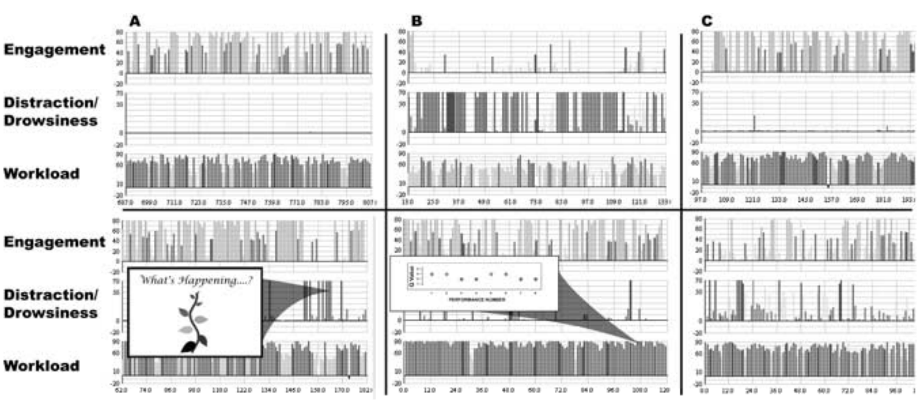


Figure 2. Dynamics of WL, D and E for Six Students on IMMEX Tasks. This figure shows 10 minute segments of the B-Alert cognitive metrics while students performed IMMEX problems.

Figure 2 illustrates the dynamics of the B-Alert EEG measures during IMMEX problem solving for six students over a ten-minute period. In each window, the top display is HE, the middle is HDT and the bottom is HWL. Each bar in the histograms represents averaged metrics at 1-second epochs.

Panels A, C and to a lesser extend F most closely represent students who were productively engaged in problem solving; workload levels were moderate and the levels were alternating with cycles of high engagement. Many cycles were associated with navigation and interpretation events (data not shown). Panel B illustrates a student who may be experiencing difficulties and might not be prepared to learn. The workload and engagement levels were low and distraction was consistently high.

The student in Panel D encountered a segment of the simulation that induced 10-15 seconds of distraction (middle row) and decreased workload and engagement. Through the data interleaving process the data that the student was looking at was retrieved, which in this case was an animation of a growing plant. Panel E shows a student who, while not distracted, appeared to be working at beyond optimal capacity with workload levels consistently near 100%. Probabilistic performance models for this student (Stevens et al, 2004, Stevens & Casillas, 2006) suggested a difficulty in developing efficient strategies on his own.

2.1. Increases in Problem Solving Skills Are Not Accompanied by Decreases in Workload or Engagement

We first measured the seconds needed to solve the first, second, and third case of *Paul's Pepperoni Pizza* (n=7) and calculated the average HWL and HE across the three performances. As shown in Table 1, the time needed to complete the task significantly decreased, however, there were no significant changes in either HWL or HE.

2.2. Students Apply Similar Workload to Similar Problems and More Workload to More Difficult Problems

Five of the students also performed 3 cases of Phyto Phyasco which is also a middle school IMMEX problem. There were no significant differences between the HWL ($0.64 \pm .05$ vs. $0.63 \pm .05$, $p = .42$) and HE ($0.51 \pm .07$, $0.51 \pm .04$, $p = .92$) across the two problem sets. Two individuals also solved the more difficult high school chemistry problem *Hazmat*. For both of these individuals the HWL was significantly greater for the three cases of *Hazmat* than for *Paul's Pepperoni Pizza*. (Subject 103: $0.76 \pm .02$ vs. $0.71 \pm .03$, $p < 0.001$; Subject 247: $0.57 \pm .02$ vs. $0.49 \pm .03$, $p < 0.005$).

The Navigation and Decision-related Events in IMMEX May Be Behaviorally Relevant

For the initial studies we divided the performance into segments related to problem framing, test selections, confirmation events where the student decides whether or not to pay the fee for the data, and closure where the student decides on the problem solution. The examples for this section were derived from one student who was performing the IMMEX mathematics problem *Paul's Pepperoni Pizza*. The student missed solving the first case, correctly solved the second case, and then missed the third case indicating that an effective strategy had not yet been formulated.

The problem framing event was defined as the period from when the Prolog first appeared on the screen until the first piece of data information is chosen. For this subject the HWL decreased from the first to the third performance ($.72 \pm .11$ vs. $.57 \pm .19$, $t = 28.7$, $p < .001$), and engagement increased $.31 \pm .30$ vs. $.49 \pm .37$ $t = 4.3$, $p < .001$). The decreased workload was similar to that observed in other subjects; the increasing HE may relate more to the student missing the problem. During the decision-making process, students often demonstrated a cycling of the B-Alert

cognitive indexes characterized by relatively high workload and low engagement which then switched to lower workload and higher engagement (Figure 3). The cycle switches were often, but not always associated with selection of new data.

Table 1. Changes in Time on Task, HWL and HE With Problem Solving Experience

Performance	Speed (seconds)	HWL	HE
1	422 ± 234	.63 ± .07	.48 ± .09
2	241 ± 126	.63 ± .08	.49 ± .08
3	136 ± 34	.65 ± .06	.47 ± .09

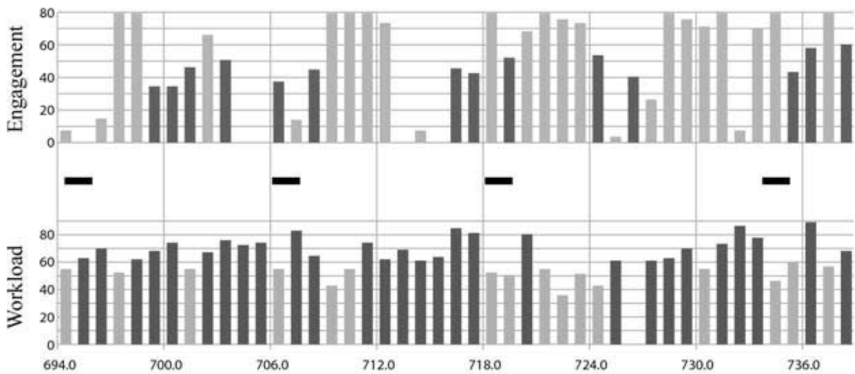
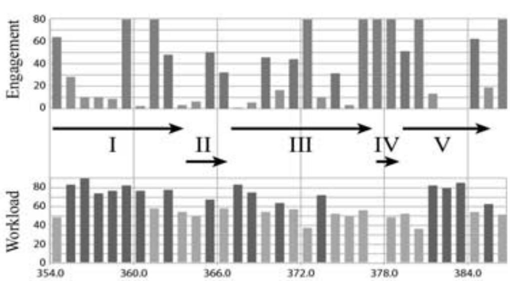


Figure 3. Fluctuations in HWL and HE during Problem Solving. The bar indicates the epochs where the student made a test selection.



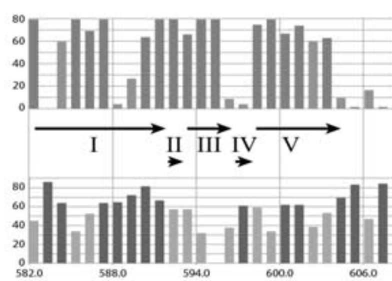


Figure 4a. Workload and Engagement Events Related to Problem Closure on the First Attempt.

Figure 4b. Workload and Engagement Events Related to Problem Closure on the Second Attempt.

The closing sequences of a problem are a complex process where first, an irrevocable decision to attempt a solution must be made. Then, a choice must be made from an extensive list of possible solutions. Finally, the choice must be confirmed. After that the student receives feedback on the success / failure of their choice; the students have two such solution attempts. The dynamics of HWL and HE for one student's first and second attempts of *Paul's Pepperoni Pizza* are shown in Fig. 4.

In the 10 seconds before deciding to solve the problem (epochs 354 – 364 (**I**)) there was HWL which decreased as the student made his decision (**II, III**). Two seconds before the student clicked on and confirmed his choice (epoch 377, **IV**) there was an increase in engagement which was maintained as the student realized that the answer was incorrect (**V**).

The workload and engagement dynamics were different on the second solution attempt. Here there was less continuous HWL in the 10 seconds leading up to the decision to solve the problem (Epochs 582- 592, (**I, II**). At epoch 593 the choice to continue was confirmed, and two seconds before making this decision engagement increased and was maintained during the selection and confirmation process. Between epochs 593 and 596 an incorrect answer was chosen and confirmed (**III, IV**). At epoch 597 the selection was made and the student learned of the incorrect answer (**V**).

2.3. Across Performance Changes in HE for Decision-Related Events.

Although there are not significant changes in HWL or HE as students develop problem solving skills, we often observed changes in these indices for the Navigation and Decision-related events across performances. For example as shown in Figure 5 (*Paul's Pepperoni Pizza*), the student correctly solved the first case, had difficulty solving the second case (missing the first try), and then failed to solve the next two cases. The overall workload and engagement levels did not change for this student across the four cases; however, there was a continual increase in the levels of HE for the submenu items where decisions were being made.

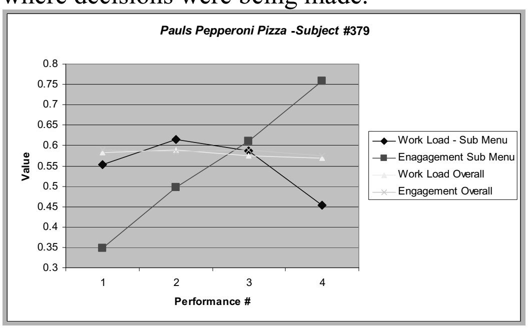


Figure 5. Changes in Overall HWL and HE and Sub Menu Linked HWL and HE Across Task Performances.

3. Discussion

In this paper we have described a web-based data acquisition architecture and event interleaving process that allows us to map EEG-derived cognitive indices to behaviorally relevant aspects of the students problem solving. An unusual feature of these studies was the application of these technologies to every-day classroom activities that are quite distinct from highly controlled laboratory tasks.

Given the anticipated differences between individual students experience and knowledge, we have focused our initial studies on comparing differences within individuals as skills are developed, rather than compare across individuals. As expected, HWL increased when students were presented with problem sets of greater difficulty. Less expected, however, was the finding that as skills increased, the levels of HWL did not decrease accordingly; suggesting significant mental commitment may be involved during strategic refinement.

By focusing the analyses around relevant problem solving events such as menu navigation and decision making, the changing dynamics of cognitive workload and engagement could be seen. By recording videos of the problem solving process and the

user on a second by second basis and interleaving them with EEG cognitive indices through log files generated by IMMEX, the majority of the HWL and HE fluctuations could be linked to observable events such as decision-making and note-taking. Initial studies suggest that decreasing workload and increasing engagement at different events of the problem solving process, such as problem framing and closure, may indicate the student experiencing difficulties and suggest bottlenecks in the learning process. These studies indicate the development of EEG-derived models of the dynamic changes in cognitive indices of workload, distraction, and engagement could be an important tool for understanding the development of problem solving skills in secondary school and university students. Long-term, such models may help target interventions to specific aspects of problem solving where the mental states of an individual reveal barriers to acquiring problem solving skills.

Supported in part by grants from the National Science Foundation (NSF-ROLE 0231995, DUE Award 0126050, ESE 9453918) and the U.S. Department of Education (R305H050052).

References

Anderson, J. R. Acquisition of cognitive skill. Psychological Review, 89, 369-406, 1982.

Berka, C., Levendowski, DJ., Cvetinovic, MM., Davis, GF., Lumicao, MN., Popovic, MV., Zivkovic, VT., Olmstead, RE., Real-Time Analysis of EEG Indices of Alertness, Cognition and Memory Acquired with a Wireless EEG Headset, International Journal of Human-Computer Interaction, 17(2), 151-170, 2004.

Berka, C., Levendowski, DJ, Ramsey, C..K., Davis, G., Lumicao, M.N., Stanney, K., Reeves, L. Harkness, Regli,, S., Tremoulet, P.D., Stibler, K., Evaluation of an EEG-Workload Model in an Aegis Simulation, Biomonitoring for Physiological and Cognitive Performance during Military Operations, edited by John Caldwell, Nancy Jo Wesentsten, Proceddings of SPIE Volume 5797, 90-99, 2005.

Ericsson, K. A. (2004). Deliberate Practice and the Acquisition and Maintenance of Expert Performance in Medicine and Related Domains. Academic Medicine. 79 (10), S70-S81.

Fabiani, M., Gratton, G. and Coles, M. G. (2000). Event-related Brain Potentials. Handbook of Psychophysiology. J. T. Cacioppo, L. G. Tassinary and G. G. Berntson. Cambridge, Cambridge University Press: 53-84.

Haider, H., and Frensch, P.A. (1996). The role of information reduction in skill acquisition. Cognitive Psychology 30: 304-337

Poythress, M., Russell, C., Siegel, S., Tremoulet, P. D., Craven, P. L., Berka, C., Levendowski, D.J., Chang, D., Baskin, A., Champney, R., Hale, K., Milham, L. Correlation between Expected Workload and EEG Indices of Cognitive Workload and Task Engagement. In Proceedings of 2nd Annual Augmented Cognition International Conference, San Francisco, CA. (In press)

Schneider, W. and Shiffrin, R. M. (1977). "Controlled and automatic human information processing I: Detection, search, and attention." Psychological Reviews 84: 1-66.

Stevens, R., & Casillas, A. (2006). Artificial Neural Networks. In R. E. Mislevy, D. M. Williamson, & I. Bejar (Eds.), Automated Scoring of Complex Tasks in Computer Based Testing: An Introduction. Lawrence Erlbaum, Mahwah, NJ.

Stevens, R., Johnson, D. F., & Soller, A. (2005). Probabilities and Predictions: Modeling the Development of Scientific Competence. Cell Biology Education 4(1): 42–57. The American Society for Cell Biology.

Stevens, R., Soller, A., Cooper, M., and Sprang, M. (2004) Modeling the Development of Problem Solving Skills in Chemistry with a Web-Based Tutor. Intelligent Tutoring Systems. J. C. Lester, R. M. Vicari, & F. Paraguaca (Eds). Springer-Verlag Berlin Heidelberg, Germany. 7th International Conference Proceedings (pp. 580-591).

Stevens, R., Wang, P., Lopo, A. (1996). Artificial neural networks can distinguish novice and expert strategies during complex problem solving. JAMIA vol. 3 Number 2 p 131-138

Stevens, R.H., & Najafi, K. (1993). Artificial Neural Networks as Adjuncts for Assessing Medical Students' Problem-solving Performances on Computer-based Simulations. Computers and Biomedical Research, 26, 172-187.

VanLehn, K., (1996). Cognitive Skill Acquisition. Annual Review. Psychology. 47: 513-539

Pedagogical Agents

This page intentionally left blank

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Tools for Authoring a Dialogue Agent that Participates in Learning Studies [1]

Pamela W. JORDAN [a,2], Brian HALL [a], Michael RINGENBERG [a], Yui CUE [b] and Carolyn ROSÉ [b]

[a] *Learning Research and Development Center, University of Pittsburgh*
[b] *LTI, Carnegie Mellon University*

Abstract. TuTalk supports the rapid development of dialogue agents for learning applications. It enables an experimenter to create a dialogue agent with either minimal or no programming and provides the infrastructure needed for testing hypotheses about dialogue. Our main goals in developing this tool were to provide 1) an authoring interface and language for setting up the domain knowledge and resources needed to support the agent and 2) a plug-and-play type of system that facilitates the integration of new modules and experimentation with different core modules. In this paper we describe the authoring tool and the usability studies that have shaped its design, the dialogue that is supported and features of the authoring language and their usage history.

Keywords. linguistics and language technology, authoring tools, intelligent tutoring, dialogue, communication

Introduction

TuTalk[3], provides a dialogue system server and authoring tool that supports the rapid development of dialogue systems to be used in learning studies. TuTalk was strongly influenced by our past tutorial dialogue research. As part of this work we created a dialogue system and authoring tools to support our studies involving knowledge construction dialogues (KCDs). A KCD is a main line of reasoning that the tutor tries to elicit from the student by a series of questions. This style of dialogue was inspired by CIRCSIM-Tutor's directed lines of reasoning [1].

TuTalk further extends the types of dialogues supported and adopts an architecture that better supports experimentation. It is intended for two classes of users; 1) non-programmers who intend to test hypotheses about dialogue in learning environments and 2) programmers who wish to measure performance differences of alternative natural language processing (NLP) modules or techniques. While the TuTalk architecture and its NLP modules are not new, a tool that is also tailored to learning applications to ease the

[1]This work was funded by NSF through award number SBE-0354420 to the Pittsburgh Science of Learning Center (PSLC) at Carnegie Mellon University and the University of Pittsburgh.

[2]Email: pjordan@pitt.edu

[3]An acronym for Tutorial Talk.

authoring process is. Thus we combine earlier work from the NLP dialogue community with that of easily authored KCDs for learning applications.

The TuTalk authoring tool enables an experimenter to set up an artificial dialogue agent for their students to interact with during an experiment. The dialogue system server can support multiple dialogue agents and a single dialogue agent can engage in dialogues simultaneously with multiple students. The authoring process consists of specifying how the domain material is to be presented to the students. Depending on how the experimenter sets up the dialogue agent, the students will engage in either agent-led or mixed-initiative dialogues and in either tutorial or conversational style dialogues. Although tutorial and conversational style dialogues are both natural language dialogues, the two are very different in character (see [1] pg. 40 for a brief discussion of the differences).

In this paper we will further describe the authoring tool and the dialogue agents supported by the server. First we provide some background on TuTalk and related dialogue systems. Next, we will give a brief overview of the the dialogues TuTalk supports and then we will discuss the features of the authoring language, their usage history and usability studies that have shaped the design of the authoring tool.

1. Background

Two representative dialogue systems from the NLP community are the DARPA Communicator architecture for spoken dialogue [17] and TrindiKit [11]. Both have modular architectures that support experimentation with dialogue system development. TrindiKit also provides a sophisticated dialogue management module that incorporates a number of dialogue theories [15] and has been used to build a variety of dialogue applications including tutorial dialogue [19]. However in all the applications created with these two architectures, none addressed the problem of non-programmers being able to create a running dialogue application. So far, only two dialogue systems in the AI in Education community have attempted to address this problem of use by non-programmers; our own Atlas system [13,16,7] and the AutoTutor system [3]. A limitation of AutoTutor is that a strategy of hint-prompt-assert is built into the system and the author cannot change this. A problem with Atlas is that it was not modular enough to be a good experimental platform and although dialogue strategy is in the hands of the author, it required programming expertise to adjust some of the general built-in dialogue behaviors; such as determining how to resume after a remediation that may have interrupted the flow of the main line of reasoning.

Atlas was used in the Why2 [16,5], Andes [13] and ITSpoke [12] physics tutoring systems. In addition, it was incorporated into the ProPL computer science tutoring system [10] and it was also used for a separate study of when during physics training dialogues are useful [9]. In total, 13 experiments in 3 science domains have made use of Atlas. All of this work successfully engaged students in natural language dialogues in that students' learn gains as measured by pre and post-tests were significant.[4] TuTalk includes all of the dialogue features of the earlier system, since all were used by the collective set of experiments, and added some new capabilities.[5] However, the dialogue agent

[4]Learning gains and the experimental conditions to which the dialogue agents were compared varied according to the experimental hypotheses being tested. Readers should see the cited papers for details on the results of some of the experiments.

[5]There is an automatic converter available that translates Atlas dialogues into the newer TuTalk format.

is re-implemented with a modular architecture to support experimentation. The architecture comprises a coordinator and a set of replaceable modules (e.g. Natural Language (NL) understanding, NL generation and Dialogue Management) and a dialogue history database. The architecture is described in detail in [8] and will not be addressed further here since our focus for this paper is the dialogue features supported.

2. An Overview of TuTalk Dialogues

The most basic dialogue that one can create with TuTalk can be represented with a finite state machine. Each state contains a single tutor turn. The arcs leaving the state correspond to all possible classifications of student turns. More complex dialogues can be created using two special arcs for calling and returning from a subdialogue. Formally, TuTalk becomes a push-down automaton (PDA) with the addition of these two special arcs because it requires that a stack be added to the finite state machine (FSM). When creating a state, the author enters the text for a tutor's turn and defines classifications for student responses. In the simplest case, a classification is defined by entering text corresponding to how the student might respond.

The NL associated with tutor turns and student responses are derived from concepts. Every concept has a concept label that is used to refer to it within an authored dialogue. A concept in Tutalk is simply a set whose elements are natural language phrases. Ideally each element of the set should represent a similar meaning. It is up to the author to decide what elements have similar meanings. For example the concept label *no* could have a set of elements that all mean a negative response to a yes/no question, such as "no", "nope" and "I guess not.". Concept definitions can be replaced by more complex representations by modifying the concept definition language and replacing the understanding and generation modules with ones that are capable of using the new concept definitions.

The dialogue manager manipulates concept labels only and relies on the NL understanding and generation modules to map between concept definitions and NL strings. Understanding is initiated when the dialogue manager requests a concept classification for an NL input. The dialogue manager supplies the student's NL input and a list of concept labels relevant to the current context to the understanding module. The understanding module returns the concept labels for the concepts that most closely match the student's complete utterance. The default understanding module provided by TuTalk uses a minimum-distance matcher to find the best concept match for an utterance but other means of mapping from NL strings to concepts can be substituted (e.g. LSA, Naive-Bayes, SVM, etc.).

Generation is initiated when the dialogue manager requests that concepts be expressed. The dialogue manager supplies the current discourse context and the labels for the concepts that are to be expressed. Generation merges together phrases appropriate to the concept definitions and the current context and requests that it be output to the student.

TuTalk's dialogue manager is implemented using the reactive planner APE [2] and decides what to express next and how to contextualize student responses. While the dialogue manager does not build-in dialogue strategies, as AutoTutor does, it does build-in some general dialogue behaviors such as refocusing after an interruption and tracking discourse obligations. Discourse obligations are obligations to respond that arise as a re-

sult of what a dialogue partner says. The dialogue manager makes use of discourse obligations to resume interrupted topics after a student initiative. Dealing with topic initiative requires more sophisticated stack manipulations than that required for a PDA since an interrupted topic may need to be pruned from the stack and the topic may span multiple stack entries. The dialogue manager also sets up tiers of concept labels (i.e. language models) for concept classification. Concepts that can potentially fulfill an obligation are in the first tier and likely concepts for initiating a new topic are in the next tier.

3. Authoring Dialogues

TuTalk's authoring language is similar to that described in [6,7]. With this language the author specifies a multi-step hierarchically-organized recipe, which is a type of plan structure defined in AI planning, for covering a topic. Recipes address high level goals and are defined as a sequence of any combination of primitive actions and non-primitive recipes [18].

First we will describe the simplest type of recipe and then we will briefly describe the following additional authoring language features; 1) recipe steps that are pointers to subrecipes, 2) responses that indicate follow-up actions, 3) automatic feedback on correctness, 4) management of the dialogue partner's focus of attention, 5) expected responses that require multiple parts to fulfill multiple discourse obligations, 6) annotating knowledge components to assess performance, 7) choosing between alternative recipes, 8) recipe steps that are to be skipped if a specified knowledge component appears in the dialogue history as an effect of some other recipe or step and 9) recipes that are to loop until a knowledge component appears in the dialogue history.

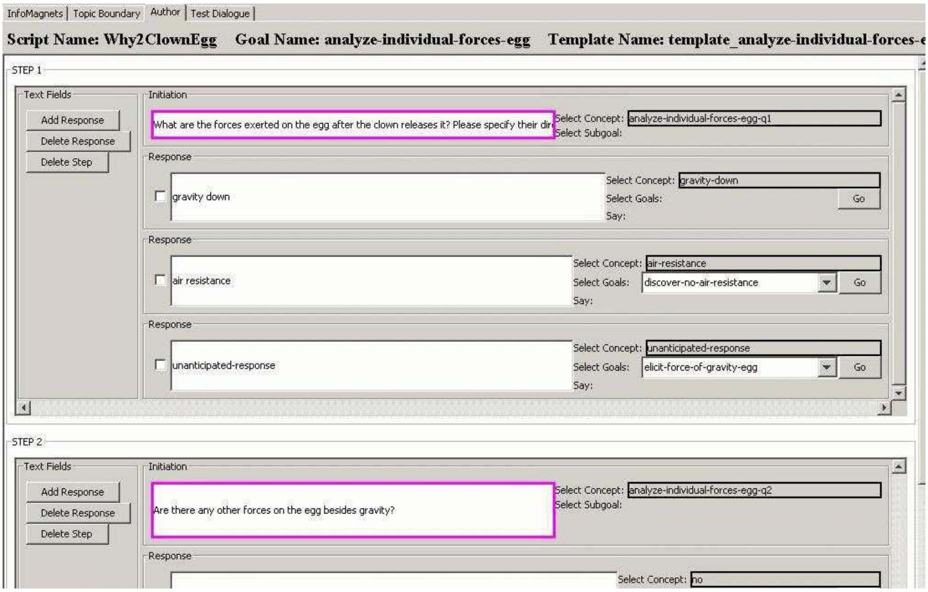


Figure 1. Authoring interface

When writing a recipe to address a topic the author creates one or more steps. The authoring interface in Figure 1 shows one full step with a second step partially visible below it. A primitive step is an initiation that can be optionally paired with a set of anticipated responses. The first step in Figure 1 shows an initiation and a set of expected responses. The specification of an initiation within a step indicates what concept to express or expect in order to achieve initiation of the step. In Figure 1, the concept labels are shown on the far right in the field *select concept* and an NL phrase associated with each is shown in the text boxes in the center. The only difference between an initiation and a response is that only one concept can be associated with an initiation. This is because an initiation corresponds to a state in a FSM and a response to a transition arc.

Subrecipes: Steps that are pointers to subrecipes enable hierarchically organized dialogue. A recipe can embed another recipe by referring to the goal name of that recipe in a step. The effect is that there is a push to the embedded recipe and when the embedded recipe completes, control returns to the parent recipe. All 13 prior experiments made use of this feature.

Follow-up actions: Responses can be authored to include follow-up actions so that vague or flawed responses can be addressed as needed. For example, *discover-no-air-resistance* in Figure 1 is the follow-up recipe that is called when the student response matches the *air-resistance* concept.

A reserved concept label of *unanticipated-response* should always be included as a possible response and an appropriate response recipe specified as with the *elicit-force-of-gravity-egg* follow-up recipe in Figure 1. The *unanticipated-response* arc is followed only if none of the expected response concepts matched. If such a response is not authored and the student's response does not match any of the others specified, then the student's discourse obligation is unfulfilled and the initiation is repeated. The assumption is that the initiation failed and that is why the student didn't respond. Multiple response actions can be indicated and are pursued in the order authored. A reserved action *abort* is now available in TuTalk and causes the parent recipe to end once control is returned to it. All 13 prior experiments used follow-up actions.

Automatic correctness feedback: The system automatically considers whether to provide correctness feedback after every student response. The *say* field on the right side of Figure 1 enables authors to provide their own correctness feedback or transition and overrides the automatic feedback for that response.

Focusing Attention: This is another automatic feature. When resuming an interruption, the dialogue manager heuristically estimates whether the student will need a reminder of the interrupted topic. To remind, it automatically repeats the turn prior to the interruption. In general dialogue this is a less than ideal way to remind the student of the interrupted topic. However, in tutorial dialogue it has the benefit of testing that a remedial interruption was effective since it was found to be a good predictor of learning outcomes [14].

Multi-part Responses: Multi-part responses are analogous to a multiple choice question in which the student needs to select multiple responses to correctly answer. In dialogue, it allows the agent to give the student partial credit and to pursue only what is wrong or missing. When authoring a multi-part response a list of the necessary and sufficient response concepts must be specified for that step. There are follow-up actions for each missing, necessary concept. The reserved concept *unanticipated-response* is se-

lected only if none of the multi-part answer concepts matches. This feature was used in 11 of 13 prior experiments. In two of these experiments it was not needed.

Annotating Knowledge Components for Performance Assessment: Initiations, responses and recipes can all be labelled with optional semantic labels. Only one semantic label per initiation, response or recipe is currently supported. The intent is that items with similar meaning are assigned the same label and can equate to a knowledge component. The dialogue system is able to react to repeated encounters of or alternative actions associated with a knowledge component. Annotated knowledge components are required for the remainder of the dialogue features described to function properly. Because recipes can be hierarchical, a subset of contiguous steps in a parent recipe can bear a knowledge component annotation. This is important because a single step in dialogue frequently does not equate to a knowledge component.

Choosing Between Alternative Recipes: This feature enables an author to create a set of alternatives for a single recipe where the recipe is associated with a targeted knowledge component and is an alternative way of covering that knowledge component. The author optionally assigns each alternative a prerequisite student performance level. The system then assesses the student's previous performance for the knowledge component in order to select an appropriate alternative. If no performance level is supplied, the default behavior is to choose the least recently used recipe when the system is to initiate discussion of the topic. This feature is relatively new and was used in 1 of the 2 experiments in which it was available. In the one experiment, 11% of the recipes had alternatives with assigned performance levels.

Optional Recipe Steps: The optional step feature allows a knowledge component, when it is about to be covered in the dialogue, to be skipped. It checks whether the semantic label on the initiation appears in the recent dialogue history and was successfully covered at that time. It the optional step is a subgoal, the semantic label for the subrecipe is checked. This feature is relatively new and was used in 1 of 2 experiments for which it was available and is now an important feature in an experiment that is currently under development. See [7] for details on optional steps and choosing between alternatives. Figure 2 shows a multi-part answer (1), an optional step (2) and follow-up actions (3).

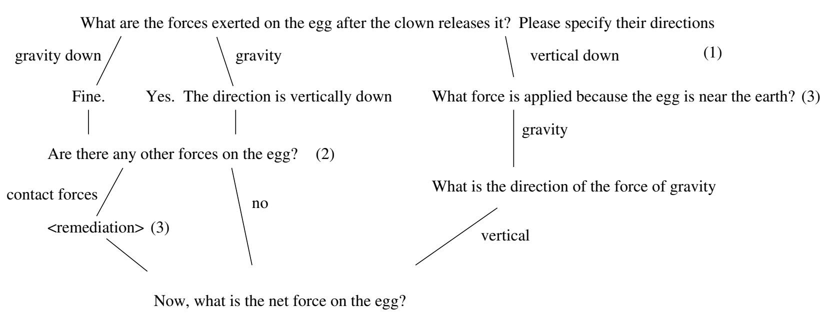


Figure 2. The paths of three Why2-Atlas students for the same recipe

Looping: The scripting language also makes use of semantic labels as termination conditions to implement looping of an entire recipe. All semantic labels included in the termination condition must appear in the dialogue history before the loop will terminate. Only *ands* of labels is supported. If an author wants only some steps in a recipe to loop, then this can be accomplished by extracting those steps into a subrecipe and setting the

loop for just that subrecipe. Currently no expiration is set for semantic labels in the case of loops. This is a new feature of TuTalk only. A previous experimenter asked about such a capability as did a prospective TuTalk user. As we see how authors use this feature we will address the issue of setting expirations on loop conditions.

4. Usability Study

The TuTalk authoring environment has been designed and developed using an iterative, user centered design methodology. An earlier paper [4] describes how our initial design for the authoring environment was based on a variety of small pilot studies and user observations in connection with earlier authoring environments. The first fully functioning prototype was then tested in Summer 2005 by 10 users with varying levels of technical expertise and no previous experience authoring dialogues. We provided the users with a short set of training materials and training tasks, which we guided them through. This required about 15 minutes. We then administered a paper-based quiz in which we tested users on the basic functionality of various interface elements in the environment. Finally, we presented the users with a second set of tasks to perform independently in the environment. A month later we tested users again to see how much knowledge they retained over time and then retrained and retested them. On average users acquired about 80% of what we exposed them to based on the immediate assessment. This dropped to 60% after one month, but increased to 85% after retraining. The most striking finding was that users seemed to have difficulty grasping the concept of the stack based structure of the authored dialogues. Observations from these studies informed the development of the environment that we released for use at the 2006 Pittsburgh Sciences of Learning Center summer school. While all participants at the summer school had some hands-on training with the TuTalk authoring environment, 9 participants used the TuTalk tools to develop an intelligent tutoring system prototype.

Past and present users of TuTalk were selected by their projects to be authors based on their domain expertise and not their knowledge of programming. Their experience ranged from none at all (2), to programming web pages (1), to people with extensive knowledge of programming. Beyond observations of use at the summer school, we have conducted in-depth observations of use in the past year and are continuing to refine the design of the authoring environment.

5. Conclusion

We have described the TuTalk dialogue agent and its authoring tool. TuTalk provides a plug-and-play type of dialogue system for learning applications that facilitates integration with new modules and experimentation with different NLP modules and an authoring language for setting up the necessary domain knowledge and resources. The 13 experiments in which the predecessor to TuTalk was used, were all successful and demonstrated that non-programmers can use it. A new experiment has demonstrated that modules within TuTalk can be replaced and supplemented. This experiment replaced the NL understanding module with a human interpretor.

We welcome new users to download and use the authoring tool.[6] It runs as a client

[6] http://andes3.lrdc.pitt.edu/TuTalk/

application and by default uses the dialogue system server that we are hosting. The dialogue system server source is also available so that researchers can set up their own servers and modify the agents it supports to meet their requirements.

References

[1] M. Evens and J. Michael. *One-on-One Tutoring by Humans and Computers*. Lawrence Erlbaum Associates, Inc., 2006.

[2] R. Freedman. Plan-based dialogue management in a physics tutor. In *Proceedings of the 6th Applied Natural Language Processing Conference*, 2000.

[3] A. C. Graesser, K. Wiemer-Hastings, P. Wiemer-Hastings, R. Kreuz, and the TRG. Autotutor: A simulation of a human tutor. *Journal of Cognitive Systems Research*, 1:35–51, 1999.

[4] G. Gweon, J. Arguello, C. Pai, R. Carey, Z. Zaiss, and C.P. Rosé. Towards a prototyping tool for behavior oriented authoring of conversational interfaces. In *Proceedings of the ACL Workshop on Educational Applications of NLP*, 2005.

[5] P. Jordan, M. Makatchev, U. Pappuswamy, K. VanLehn, and P. Albacete. A natural language tutorial dialogue system for physics. In *Proceedings of the 19th International FLAIRS conference*, 2006.

[6] P. Jordan, C. Rosé, and K. VanLehn. Tools for authoring tutorial dialogue knowledge. In *Proceedings of AI in Education 2001 Conference*, 2001.

[7] P. W. Jordan, P. Albacete, and K. VanLehn. Taking control of redundancy in scripted tutorial dialogue. In *Proceedings of the Int. Conference on Artificial Intelligence in Education, AIED2005*. IOS Press, 2005.

[8] P. W. Jordan, M. Ringenberg, and B. Hall. Rapidly developing dialogue systems that support learning studies. In *Proceedings of ITS06 Workshop on Teaching with Robots, Agents, and NLP*, 2006.

[9] S. Katz, J. Connelly, and C. L. Wilson. When should dialogues in a scaffolded learning environment take place? In *Proceedings of EdMedia 2005*, volume 65, 2005.

[10] H. C. Lane and K. VanLehn. Teaching the tacit knowledge of programming to novices with natural language tutoring. *Computer Science Education*, 15(3):183–201, September 2005.

[11] S. Larsson and D. Traum. Information state and dialogue management in the trindi dialogue move engine toolkit. *Natural Language Engineering Special Issue on Best Practice in Spoken Language Dialogue Systems Engineering*, pages 323–340, 2000.

[12] D. J. Litman and K. Forbes-Riley. Predicting student emotions in computer-human tutoring dialogues. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL)*, Barcelona, Spain, July 2004.

[13] C. Rosé, P. Jordan, M. Ringenberg, S. Siler, K. VanLehn, and A. Weinstein. Interactive conceptual tutoring in atlas-andes. In *Proceedings of AI in Education 2001 Conference*, 2001.

[14] M. Rotaru and D. Litman. Exploiting discourse structure for spoken dialogue performance analysis. In *Proceedings of the Empiricial Methods in Natural Language Processing Conference*, 2006.

[15] D. Traum and J. Allen. Discourse obligations in dialogue processing. In *ACL94, Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*, pages 1–8, 1994.

[16] K. VanLehn, P. Jordan, C. Rosé, D. Bhembe, M. Böttner, A. Gaydos, M. Makatchev, U. Pappuswamy, M. Ringenberg, A. Roque, S. Siler, and R. Srivastava. The architecture of Why2-Atlas: A coach for qualitative physics essay writing. In *Proceedings of Intelligent Tutoring Systems Conference*, volume 2363 of *LNCS*, pages 158–167. Springer, 2002.

[17] M. Walker, A. Rudnicky, R. Prasad, J Aberdeen, E. Bratt, J. Garofolo, H. Hastie, A. Le, B. Pellom, A. Potamianos, R. Passonneau, S. Roukos, G. Sanders, S. Seneff, and D. Stallard. DARPA communicator: Cross-system results ofr the 2001 evaluation. In *Proceedings of International Conference on Spoken Language Processing (ICSLP)*, 2002.

[18] R. Michael Young, Martha E. Pollack, and Johanna D. Moore. Decomposition and causality in partial order planning. In *Second International Conference on Artificial Intelligence and Planning Systems*, 1994. Also Technical Report 94-1, Intelligent Systems Program, University of Pittsburgh.

[19] C. Zinn, J. Moore, and M. Core. Intelligent information presentation for tutoring systems. In O. Stock, editor, *Multimodal Intelligent Information Presentation*. Springer, 2005.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Changing Attitudes and Performance with Computer-generated Social Models

Rinat B. ROSENBERG-KIMA[1], E. Ashby PLANT[2], Amy L. BAYLOR[1], Celeste E. DOERR[2]

Center for Research of Innovative Technologies for Learning (RITL) [1], Florida State University
Innovation Park, 2000 Levy Avenue, Bldg. A, Suite 320
Tallahassee, FL 32310-2735
Department of Psychology[2], Florida State University
Tallahassee, FL 32306

Abstract. Women's under-representation in fields such as engineering may result in part from female students' negative beliefs regarding these fields and their low self-efficacy for these fields. Empirical evidence indicates that computer-generated interface agents are effective in influencing students' interest, motivation, attitudes, and self-efficacy. Hence, in this experimental study, we investigated the potential of interface agents to serve as effective social models for changing attitudes regarding the utility of math and the hard sciences and self-efficacy for these fields. 113 middle-school students interacted with either a female or a male computer-generated interface agent or they did not interact with an interface agent. The findings from this study indicate that interface agents may be used effectively as social models for influencing middle school students' attitudes and beliefs about mathematics and the hard sciences and their mathematical ability. Nevertheless, the efficacy of the agent depended on the characteristics of the agent with the female agent tending to be the most effective regardless of the subject gender.

Keywords. Anthropomorphic interface agents, persuasion, attitude change, computer-based social modeling

1. Introduction

In this experimental study, we investigated the potential of computer-generated agents to serve as effective social models in math and the hard sciences for middle-school students. Participants interacted with either a female or a male computer-generated social model or they did not interact with an interface agent. We investigated the influence of the social interface agents on students' attitude regarding the utility of math and the hard sciences and their self-efficacy for these fields. A math test assessed the implications of the exposure to the agent for students' mathematical ability.

1.1. Gender Differences in Sciences Related Fields

Although there is an evident increase in women's inclusion and success in professions that were traditionally occupied primarily by men, they remain under-represented in the fields such as engineering that require skills in mathematics and the hard sciences [1]. Women's under-representation in fields such as engineering may result in part from female students' negative beliefs regarding these fields and their low self-efficacy with regard to the abilities required for these fields [2].

Self-efficacy refers to the belief that one is competent to meet situational demands [e.g., 3, 4]. Research shows that female engineering students tend to perceive themselves as less competent than their male peers [1]. Those negative self-efficacy beliefs begin early. Girls as young as elementary age tend to underestimate their math ability, even though their actual performance may be equivalent to that of same-aged boys [5-7] and their computation skills are even slightly better in elementary school and middle school [8].

In addition to low self-efficacy, women and girls possess unproductive beliefs about math and the hard sciences, sex-typing science as a masculine pursuit [e.g., 9], and negating the utility of mathematics [6]. Such negative responses to math and the hard sciences may make young women less likely to pursue these areas and may lead them to stop taking classes in these fields from an early age. However, if young women are reached at a young age, it may be possible to change these unproductive beliefs about mathematics and the hard sciences.

1.2. Interface Agents as Social Models

According to Bandura [e.g., 10, 11], much of our learning derives from vicarious experiences. Social modeling of behaviors enables us to learn new behaviors, strengthens or diminishes previously learned behaviors, and reminds us to perform behaviors about which we had forgotten. Social models can also influence people's attitudes [12]. Observing a social model who is similar to the self perform a behavior in a domain provides people with information relevant to their likely self-efficacy in that domain [13]. Therefore, exposure to social models, particularly young, female models who are in the mathematic and the hard sciences fields, may be helpful for improving young women's attitudes and self efficacy regarding math and the hard sciences.

Providing young, female social models in math and the hard sciences to students may be problematic because it would contribute to the already burdensome workloads faced by women in nontraditional fields [14]. In addition, there may exist a possible lack of fit between available social models and the needs of individual students. Therefore, it would be useful to find alternative mechanisms for exposing young women to social models in the mathematics and hard sciences fields.

Interface agents are animated characters that provide teaching or mentoring within a computer-based learning environment. Empirical evidence indicates that interface agents are effective in promoting learning [e.g., 15, 16] and also metacognition [17]. Interface agents have also been found to positively influence student interest, motivation [18-21], and attitudes [22, 23]. Interface agents can also influence self-efficacy [22]. Thus, it may be possible to use interface agents as social models.

Extensive research has demonstrated that people tend to apply human social rules to computer technologies. Further, young women are particularly influenced by the communication and relational aspect of agents and may be more influenced by them than

males [21, 24]. Therefore, interface agents, as simulated social models, may be particularly helpful in affecting young female's attitudes and self-efficacy with respect to math and sciences.

1.3. Purpose of Study

The purpose of the current study was to test whether computer-generated agents can serve as effective social models for math and the hard sciences for middle-school students. Participants interacted with either a female or a male computer-generated interface agent or they did not interact with an interface agent. A questionnaire assessed participants' attitudes regarding the utility of math and the hard sciences and their self-efficacy for these fields. A math test assessed the implications of the exposure to the model for students' mathematical ability.

Participants who interacted with an interface agent were expected to indicate more positive attitudes regarding the utility of math and science and greater self-efficacy for math than participants who did not interact with an interface agent. These findings were expected to be particularly strong if the interface agent matched the gender of the participant (i.e., girls would be more influenced by the female interface agent).

2. Method

2.1. Participantsd

Sixty-five females (age M = 13.78, SD = 1.15), and 48 males (age M = 13.87, SD = 1.07) middle-school students participated during a regular scheduled class session. Of the participants, 53.1% were Caucasian, 28.3% were African-American, 1.8% were Asian/Asian American, 8% were Hispanic/Latino, 5.3% were multiracial, 0.9% defined their ethnicity as "other", and 2.7% did not report their ethnicity.

2.2. Research Design and Independent Variables

The study employed a 2 (Participant Gender: male vs. female) x 3 (Interface agent: female vs. male vs. no interface agent) between subjects factorial design. Participants interacted with either a female or a male computer-generated interface agent or they did not interact with an interface agent. All participants filled a questionnaire that was designed to assess their attitudes and self-efficacy regarding engineering-related fields and their mathematical ability.

In previous related work, we found that attractive agents were more influential as social models for engineering [21]. In addition, we found that among the attractive agents, the young and cool were the most influential [22]. Consequently, the agents for the current study were designed to be young (~25 years), cool (as manipulated by the agent's clothing and hairstyle), and attractive (as manipulated by the agent's facial features) and varied in their gender. Previous testing of the agents confirmed that participants perceived them as young, cool, and attractive [22]. The agents (see Figure 1) were created in Poser©.

Figure 1. Female and male young, attractive, and cool agents.

Audio files of human female and male were synchronized with the agents using Mimic2Pro© to create lip-synching and emotional expressions. Several deictic gestures (e.g., pointing a finger) were also included. These gestures were identical for all agents. A fully integrated environment was created using Flash MX Professional©.

2.3. Dependent Variables

Because mathematics and the hard sciences (e.g., chemistry, physics) are strongly related to the field of engineering and are important prerequisites for many types of engineering, we measured participants' attitudes and beliefs regarding these engineering-related fields. The dependent variables were perceived utility, self-efficacy, and career interest in these fields. In addition, we measured the participants' math ability.

The dependent variables were all measured using a 7-point, Likert-type scale. Items for these scales were duplicated, so that half of the items in each scale referred to math and half referred to the hard sciences. Eight items assessed participants' beliefs about the utility of engineering ($\alpha = .87$; e.g., "I would have many good career opportunities if I was a hard science major"). Ten items assessed students' self-efficacy in engineering-related fields ($\alpha = .87$; e.g., "I have always done well in math"). Two items assessed students' interest in having a career in engineering-related fields ($\alpha = .62$; "How likely would you be to take a job in a hard sciences related field?"). Finally, eight multiple-choice questions assessed the participants' math ability.

2.4. Research Environment

Each student was randomly assigned to one of the three conditions (Interface agent: female vs. male vs. no interface agent). In the first two conditions, the assigned agent (set in a coffee shop location) introduced itself and provided a twenty-minute narrative about four female engineers, followed by five benefits of engineering careers. Both the female and the male agents provided exactly the same narrative. This script was validated as effective in Baylor and Plant [21]. Periodically, the participants interacted with the agent to continue the presentation.

2.5. Procedure

The experiment was conducted in a regularly scheduled class session where students accessed the online module through a web-browser (see Figure 2 for screen-shot).

Following completion (or immediately in the case of the control group), participants answered the online post-survey questions.

Figure 2. Sample screenshot.

3. Results

To determine the impact of the interface agents and the different impact of the female and male agents, a series of 2 (participants' gender: female vs. male) x 3 (Interface agent condition: female vs. male vs. no agent) between-groups ANOVAs were performed on each of the key dependent variables. Post-hoc tests were conducted where relevant (see table 1).

The analysis for utility revealed a significant main effect of interface agent condition, $F(1, 112)=3.26$, $p < .05$. Participants who interacted with a female computer-generated interface agent expressed significantly more positive beliefs about math and the hard sciences' utility than participants who did not interact with an agent ($d = .64$). The male agent condition fell in between and did not significantly differ from either the no agent or female agent condition.

The analysis for self efficacy revealed a significant main effect of interface agent condition, $F(1, 112)=4.92$, $p < .01$. Participants who interacted with a computer-generated interface agent expressed significantly greater self-efficacy for their performance in engineering-related fields than participants who did not interact with an agent (female vs. no agent, $d = .79$; male vs. no agent, $d = .68$). There was no significant difference between the female agent and the male agent conditions.

The analysis for career interest revealed a significant main effect of interface agent condition, $F(1, 112)=4.71$, $p < .01$. Participants who interacted with a female computer-generated interface agent expressed significantly greater interest in engineering-related careers than participants who did not interact with an agent, ($d = .62$) and than participants who interact with the male agent ($d = .66$). There was no significant difference between the no agent condition and the male condition.

Finally, the analysis for math performance also revealed a significant main effect of interface agent condition, $F(1, 112)=4.09$, $p < .05$. Participants' performance on the math test was significantly higher when they interacted with a female computer-generated interface agent than when they did not interact with an agent, ($d = .66$). Participants who interacted with the female agent also performed marginally better on the math test ($p = .06$) than those who interacted with the male agent ($d = .40$). There was no significant difference between the no agent condition and the male condition.

Table 1. Effectiveness of Interface Agents

DV	F	Condition	M	SD
Utility	3.26*	No agent	4.39 [a]	1.64
		Female agent	5.33 [b]	1.28
		Male agent	4.88 [ab]	1.29
Self efficacy	4.92**	No agent	3.44 [a]	.73
		Female agent	4.02 [b]	.74
		Male agent	3.93 [b]	.70
Career interest	4.71**	No agent	2.70 [a]	1.24
		Female agent	3.40 [b]	1.01
		Male agent	2.67 [a]	1.21
Math test scores	4.09*	No agent	3.55 [a]	1.73
		Female agent	4.65 [b]	1.62
		Male agent	4.00 [ab]	1.65

Note: Significance of F values is identified by * at the .05 level and by ** at the .01 level. Differing superscripts indicate significant differences between means on post hoc tests.

4. Discussion

The current work examined whether computer-generated interface agents can serve as effective social models for math and the hard sciences for middle-school students. Participants interacted with either a female or a male computer-generated interface agent or they did not interact with an interface agent. Supporting our hypotheses, the findings from this study indicate that interface agents may be used effectively as social models for influencing middle school students' attitudes and beliefs about mathematics and the hard sciences as well as their actual mathematical ability. However, the effectiveness of the agent depended on the characteristics of the agent with the female agent tending to be the most effective regardless of the subject gender.

Specifically, the female agent was significantly better than the no agent condition in influencing both females and males' beliefs about the utility of math and the hard sciences. In addition, females and males students who interacted with the female model performed significantly better on the math test. For both utility and mathematical performance, the male agent fell in between and was not significantly different compared to the female condition and the no agent condition. In addition, for both the females and males' interest in engineering-related careers, the female agent resulted in significantly more interest than both the no agent and the male agent.

However, for the students' self-efficacy, having an agent was better than not being exposed to an agent regardless of the model gender. That is, both the male and the female models were significantly better than the no model in increasing females and males' self-efficacy with regard to their performance in engineering-related fields.

Contrary to our hypothesis, the interface agents influenced females and males students in the same way, with the female agent being more effective for most of the outcome measures. The effectiveness of the female model may be due to middle-school students' frequent contact with female teachers. This contact with female teachers may predispose them to view female social models as informative. It is also possible that the female model's positive impact was due to different factors for female and male students. The female students may have identified with the female model and seeing the female model may have led the young women to view these disciplines as welcoming to women. The male students, in contrast, may have been persuaded by the attractive female model that math and science are appealing disciplines where they might meet desirable women, a belief that is generally not a part of the stereotype of these disciplines.

Female computer-generated social models were especially effective for enhancing students' mathematical performance. For female students, exposure to the successful, female social model may have mitigated stereotype threat [e.g., 25, 26]. Stereotype threat is a disruptive concern that one may confirm a negative stereotype about one's social group which can result in impaired performance in relevant situations [26, 27]. Mitigation of stereotype threat is unlikely to explain male students' enhanced performance. However, it is possible that the males associate mathematics with attractive women after receiving the message from the attractive female agent, and this may have increased their motivation to do well in mathematics.

Although the current work provides some evidence of the efficacy of interface agents to improve responses to mathematics and the hard sciences, there were some limitations of the current work. One limitation of this study was that the students who were in the no agent group did not get a substitute treatment prior to filling the survey and answering the mathematical test. Significant results may be due to the existence of a treatment and not the interface agent itself. Nevertheless, for most of the dependent variables, only the female agent was significantly better than the no agent condition, thus, the effects cannot be contributed solely to the existence of a treatment. Another possible limitation was the lack of a pre-test control. A pre-test in this case would have exposed the participants to the content and purpose of the persuasive message, thereby negating the purpose of the study (e.g., through expectancy or demand effects). Instead, we utilized a randomly-assigned, between-groups, experimental design. By randomly assigning participants to experimental conditions, we were able to examine differences between the experimental groups without cueing participants to the content of the persuasive message.

In addition, in the current study, we only manipulated the agents' gender. Other characteristics of the agent such as voice (e.g., computer-generated vs. human, tone, accent), persona (e.g., its personality, affective state), and animation (e.g., emotional expressiveness, deictic gestures) were held constant in this study. Future studies should consider such design elements and how they may potentially interact with the agent's gender.

To summarize, we found that computer-generated people can serve as effective social models for math and the hard sciences for middle-school students. In this study we assessed the students' attitudes and beliefs about mathematics and the hard sciences and their mathematical ability using a survey instrument. Of future interest will be measuring how the models actually affect students' choices with regard to their classes and future career.

References

[1] Goodman, I.F., et al., *Final report of the women's experiences in college engineering project*. 2002, Cambridge, MA: Goodman Research Group, Inc. .

[2] Shashaani, L., *Gender differences in computer attitudes and use among college students.* . Journal of Educational Computing Research, 1997. **16**: p. 37-51.

[3] Bandura, A., *Self-efficacy: Toward a unifying theory of behavioral change.* . Psychological Review, 1977. **84**: p. 191-215.

[4] Wood, R. and A. Bandura, *Impact of conceptions of ability on self-regulatory mechanisms and complex decision making.* Journal of Personality and Social Psychology, 1989. **56**: p. 407-415.

[5] Eccles, J.S., *Gender roles and women's achievement-related decisions.* Psychology of Women Quarterly, 1987. **11**: p. 135-172.

[6] Eccles, J.S., *Understanding women's educational and occupational choices: Applying the Eccles et al. model of achievement related choices.* Psychology of Women Quarterly, 1994. **18**: p. 585-609.

[7] Seymour, E. and N.M. Hewitt, *Talking about Leaving – Why Undergraduates Leave the Sciences.* 1997, Boulder: Westview Press.

[8] Hyde, J.S., E. Fennema, and S.J. Lamon, *Gender Differences in Mathematics Performance: A Meta-Analysis.* Psychological Bulletin, 1990. **107**(2): p. 139-155.

[9] Hughes, W.J., *Gender Attributions of Science and Academic Attributes: An Examination of Undergraduate Science, Mathematics, and Technology Majors.* Journal of Women and Minorities in Science and Engineering, 2002. **8**(1): p. 53-65.

[10] Bandura, A., *Self-efficacy: The exercise of control.* 1997, New York: W.H. Freeman and Company.

[11] Bandura, A., *Social Foundations of Thought and Action: A Social Cognitive Theory.* . 1986, Englewood Cliffs, N.J.: Prentice-Hall.

[12] Goethals, G.R. and R.E. Nelson, *Similarity in the influence process: The belief-value distinction.* Journal of Personality and Social Psychology, 1973. **25**(1): p. 117-122.

[13] Bandura, A. and D.H. Schunk, *Cultivating competence, self-efficacy, and intrinsic interest through proximal self-motivation.* Journal of Personality and Social Psychology, 1981. **41**(3): p. 586-598.

[14] Hersh, M., *The changing position of women in engineering worldwide.* Ieee Transactions on Engineering Management, 2000. **47**(3): p. 345-359.

[15] Atkinson, R.K., *Optimizing learning from examples using animated pedagogical agents.* Journal of Educational Psychology, 2002. **94**(2): p. 416-427.

[16] Moreno, R., et al., *The case for social agency in computer-based teaching: Do students learn more deeply when they interact with animated pedagogical agents?* Cognition and Instruction, 2001. **19**(2): p. 177-213.

[17] Baylor, A.L., *Expanding preservice teachers' metacognitive awareness of instructional planning through pedagogical agents.* Educational Technology, Research & Development, 2002. **50**: p. 5-22.

[18] Baylor, A.L., *Agent-based learning environments for investigating teaching and learning.* Journal of Educational Computing Research, 2002. **26**: p. 249-270.

[19] Baylor, A.L. and Y. Kim, *Simulating instructional roles through pedagogical agents.* International Journal of Artificial Intelligence in Education, 2005. **15**(1): p. 95-115.

[20] Kim, Y. and A.L. Baylor, *Pedagogical agents as learning companions: The role of agent competency and type of interaction.* Educational Technology Research & Development, 2006. **54**(3): p. 223-243.

[21] Baylor, A.L. and E.A. Plant, eds. *Pedagogical agents as social models for engineering: The influence of appearance on female choice.* Artificial intelligence in education: Supporting Learning through intelligent and socially informed technology ed. C.K. Looi, et al. Vol. 125. 2005, IOS Press. 65-72.

[22] Baylor, A.L., E.A. Plant, and C.E. Doerr, *Pedagogical Agents as Social Models for Female Students: The Effects of Agent Appearance on Women's Attitudes and Beliefs.* . 2007: under review.

[23] Baylor, A.L., R.B. Rosenberg-Kima, and E.A. Plant. *Interface Agents as Social Models: The Impact of Appearance on Females' Attitude Toward Engineering.* . in *CHI 2006 (Conference on Human Factors in Computing Systems).* 2006. Montreal, Canada.

[24] Baylor, A.L. *The Impact of Pedagogical Agent Image on Affective Outcomes*. in *International Conference on Intelligent User Interfaces*. 2005. San Diego, CA.

[25] Steele, C.M., *A threat in the air: How stereotypes shape intellectual identity and performance.* American Psychologist, 1997. **52**: p. 613–629.

[26] Steele, C.M. and J. Aronson, *Stereotype threat and the intellectual test performance of African Americans.* Journal of Personality and Social Psychology, 1995. **69**: p. 797–811.

[27] McIntyer, B., R.M. Paulson, and C.G. Lord, *Alleviating women's mathematics stereotype threat through salience of group achievements.* . Journal of Experimental Social Psychology, 2003. **39**: p. 83-90.

Pedagogical Agents Trying on a Caring Mentor Role

Konstantin ZAKHAROV[a], Antonija MITROVIC[a], Lucy JOHNSTON[b]
[a] *Intelligent Computer Tutoring Group,* [b] *Department of Psychology*
University of Canterbury, Christchurch, New Zealand
{konstantin.zakharov, tanja.mitrovic, lucy.johnston}@canterbury.ac.nz

Abstract: We describe the design and evaluation of an affective pedagogical agent persona for Intelligent Tutoring Systems. The goal of our research was to develop an agent embodying a persona of a caring mentor interested in the learner's progress. The agent's behaviour is guided by a set of rules that are triggered by the states of the session history. Four agents were integrated with EER-Tutor for a formative evaluation study. The mentor persona secured strong rapport with the users; the audible narration was seen as a strong feature of the agents.

Introduction

The semantic component of social interaction, most frequently represented as speech, is often underpinned by the affective component, which can be expressed through speech and non-verbal displays such as gestures, posture, facial expression and eye gaze [10]. People have a propensity to transfer the social view of interaction onto their interaction with electronic media. As described by Reeves and Nass [18], people tend to view electronic media in a social way, as if they were other people. However, computers since their early days have been implicitly designed without awareness of the affective communication channel. Computers respond to people as if they too were computers, thus forcing people to adjust to computer protocols and interact on a sub-human level, which contradicts the main assumption guiding Human-Computer Interaction (HCI) research: *"People should not have to change radically to 'fit in with the system' – the system should be designed to match their requirements"* [17].

This inherent lack of affective fit is particularly significant in the area of Intelligent Tutoring Systems (ITS), because research suggests a strong interaction between cognitive and affective processes in human mind. For example, there is a reliable correlation between one's emotional state, memory capacity and motivation [6, 18]. Consequently, a failure to recognise affective processes might impose a serious limitation on interaction types which are fundamentally social in nature, because the affective component is often considered as important as semantic component [3]. Without considering the interaction between the cognitive and affective processes ubiquitous in human interaction, educational systems might never approach their full potential.

HCI does acknowledge the need to avoid negative affective states, such as frustration; common solutions to avoid user frustration include either (a) trying to determine and fix the problem causing the negative feelings, and/or (b) pre-emptively trying to avoid the problem from happening in the first place [7]. However, these approaches have a limited application in ITSs because of the fundamental differences between

general HCI and educational protocols. Affective considerations in ITSs are more complex, because learning from a computer is not just about ease of use. It would be unrealistic to expect students to stop making errors during learning. Learning can be frustrating and difficult because it requires exposing learners' errors in thinking and gaps in their knowledge. Theory of Learning from Performance Errors [15] suggests that errors are an inseparable component of learning. Error detection followed by error correction, in fact, is vital for the improvement of future performance.

Recent research suggests using Affective Pedagogical Agents (APAs) in ITSs as a medium for delivering feedback to the users [8, 12]. This research draws on Allport's [1] classic definition of social psychology: *"The scientific investigation of how the thoughts, feelings and behaviours of individuals are influenced by the actual, imagined or implied presence of others"*; in accordance with this statement APAs are known to enhance the social view of interaction with ITSs. This undermines the idea that computers are merely neutral tools and emphasises the importance of the social relationships that can develop between a computer and a learner [14]. A better understanding of these relationships is essential to building smarter tools for learning.

We start by presenting our work on developing an affective persona for agents integrated with EER-Tutor, an ITS for developing Enhanced Entity-Relationship (EER) modelling skills [19, 21]. Section 2 outlines the experiment aimed at assessing learners' perception, expectations and response to the agents, and details the experimental results. Section 3 presents the conclusions and discussion of our research.

1. Developing Affective Pedagogical Agents

When designing the persona, we had to establish what kind of affective response an agent should provide in order to support the learner's determination in the face of the inevitable stress, anxiety and frustration involved in learning. One simple rule of thumb suggested by Bickmore [3] is to apply what has been found appropriate for human-to-human interaction (HHI) to the design of educational HCI. People use a variety of methods to help manage their emotions, such as interacting with media and/or other people, engaging in sports or work, meditating or praying, using positive thinking, and consuming foods and substances such as alcohol. Klein et al. [9] identify two types of support for emotion regulation. First, passive support is used to manipulate moods without necessarily discussing emotions themselves. Media, activities, food and other substances fall into this category. In contrast, active support occurs when people discuss or otherwise directly address their emotions, as a means of managing them.

Out of possible instructional roles we chose the mentor role, because of positive influence on learning demonstrated in previous research [2]. At the same time, from the affective interaction point of view, passive affective support intuitively seems as adequately congruent with the role of a mentor. Consequently, we tried to create an agent with a persona of a mentor interested in learner's progress. We wanted the agent to acknowledge the leaner's emotions indirectly through its emotional appearance, while trying to keep the learner focused on the task at hand. Thus we designed an agent which expresses solidarity with the user – it will cheer with the users' success, be sympathetic when there are difficulties and keep company to the user in neutral situations. Similar agent's behaviour was earlier adopted in the work of Lester et al. [11].

Human mentors can pick up a lot of clues from non-verbal communication; with varying degrees of depth, the mentor is always aware of the learner's affective state and

cognitive state. In combination, these two factors allow a human mentor to choose an appropriate affective response at each step. While in our case, the agent does not have a way of determining users' affective state, prior research shows that in a learning context affective state can be indexed on the basis of cognitive state [4]. Cognitive Theory of Emotions states that the valence of one's emotional reaction depends on the desirability of the situation [16]. Thus we can assume that continuous lack of cognitive progress will be accompanied by a negative affective state, because the user will be dissatisfied with the state of the current task. Conversely, good progress will result in a positive affective state. In our approach, we rely on a simplified emotional model described by a single dimension – affective valence.

EER-Tutor maintains student models; the state of student model can be used to index the student's affective states. In our agents, however, affective logic does not directly rely on the student model. Instead, the logic relies on session history, which includes a wide variety of user actions. The rationale behind this approach is twofold. First, most user actions, such as login, problem selection and so on, may require a response or acknowledgement from the agent; in many cases such responses would not have to carry much affective load, although failing to respond in a similar HHI situation could be interpreted as lack of attention and care. Second, under certain circumstances seemingly neutral actions might indicate changes in the users' affective state and thus should be addressed by the agent. For example, repeated submissions of the same solution to a problem might indicate a negative affective state, such as frustration. In general, we assume that any repeating actions should not go unnoticed.

1.1. Agents' Appearance

EER-Tutor [19, 21] is a web-based ITS whose server carries the application load which includes running the constraint-based modelling engine, applying pedagogical logic, maintaining student models and hosting EER-Tutor curriculum. The client is implemented as a set of AllegroServe[1] dynamic HTML pages and a Java applet providing drawing tools for creating solutions; the interface only accepts user input and delivers feedback. Similarly, the agents' implementation is also split between the server and client. The server carries the agents' affective logic and controls the agents' behaviour, while on the client-side the agent appears to the users as a "talking head" with an upper body, embedded in EER-Tutor's work-space. The agent figures have been designed with the help of PeoplePutty[2] toolkit; the web browser displays the agent with Haptek[3] player plug-in. Haptek's character affective appearance is controlled by a number of parameters called switches. Some switches, for example, control lips and eyebrows positions. Haptek characters communicate with the server through AJAX[4] requests. The communication process revolves around retrieving updated values for the switches along with appropriate narration fragments. Figure 1 shows EER-Tutor workspace with a male agent above the feedback pane. The screenshot shows the state immediately after a solution submission. In this case, the submitted solution was incorrect: the user mistakenly defined the *Chapter_number* attribute as a key attribute, instead of making it a partial key attribute. The erroneous diagram component, the *Chapter_number* key

[1] http://allegroserve.sourceforge.net – AllegroServe web application server
[2] http://www.haptek.com/products/peopleputty/ — Haptek's PeoplePutty SDK
[3] http://www.haptek.com/ — PeoplePutty is a product of Haptek
[4] http://www.adaptivepath.com/publications/essays/archives/000385.php

attribute, is shown in red. In general, the error messages presented by the agent refer the user to the errors in the diagram; in our example the first error message says: *A weak entity type must have a partial key. The highlighted entity type has no partial key*. This approach to engaging the user in active information processing is supported by Mayer's first principle of multi-media design, *Multimedia effect for transfer*, which states that providing the learner with a narration results in better learning when it is supplemented by corresponding visual aids [13].

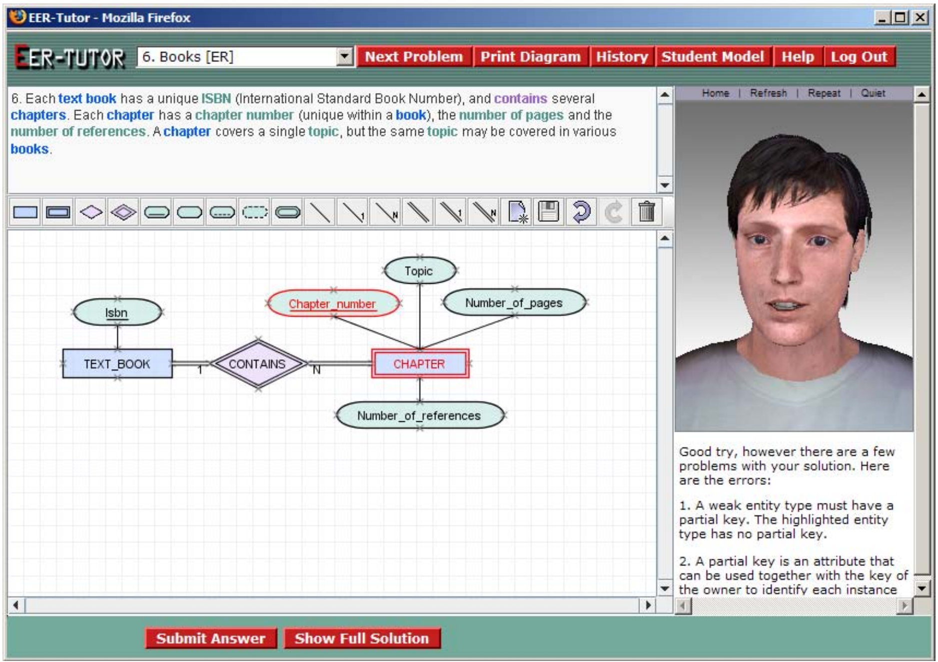


Figure 1. The view of EER-Tutor's work-space with the male agent

Figure 2. The agents designed with the PeoplePutty toolkit.

Along with feedback in audible form, the user is also presented with the most recent feedback messages in the textual form. Even though the sixth principle of multi-media design, *Redundancy effect for transfer*, states that such redundancy negatively

affects learning [13], we consider that the EER-Tutor context justifies such redundancy, because the complexity of the domain knowledge and abundance of information may be difficult to process without textual representation. For example, when the user makes several mistakes, the corresponding messages may help the user to stay focused on the task instead of wasting efforts on remembering multiple feedback messages.

Haptek figures rely on Microsoft's Speech Application Programming Interface (SAPI), SAPI4 and SAPI5-compatible Text-to-Speech (TTS) engines to display realistic mouth movements while producing verbal narrations. In order to enhance the agents' realism, we obtained two reportedly higher quality Cepstral[5] voices – one male and one female, instead of using the default TTS voices supplied with Windows XP. Figure 2 shows the four agents we created – two male and two female characters. We chose a generic youthful appearance in order to remain consistency with agent's role of a mentor' since the mentoring group is traditionally represented by younger people.

1.2. Agents' Behaviour

EER-Tutor is driven entirely by user-generated interface events, such as submissions of solutions. Every interface event results in a corresponding request type being sent to the server. When the server processes the request and the client receives the response, the agent control script requests the server for the updates of the agents' emotional display and verbal feedback to be returned in response to the users' most recent action. At this stage the agent's behaviour rules, described later in this Section, are matched against the session history. On completion of this process, the server sends the response defined by the selected rule. In this way, even though the Haptek character is completely unaware of the users' actions and student model, to the user it appears that the agent is actively observing their actions.

The agent control script also runs an uninterrupted loop (repeating at the 1.5s intervals) continuously querying the server for affective appearance and narration updates even in the absence of interface events. While user actions may colour the agent's affective appearance, the agent is capable of emotional self-regulation mimicking human behaviour. The agent's affective state tends to gradually return to neutral, so if the agent momentarily may appear very upset or happy, after a few minutes the agent inevitably "settles down" even in the absence of affect-provoking actions of the user.

The agent's persona is controlled by a set of rules, each of which corresponds to a unique state of the session history and produces a certain reaction from the agent; every rule consists of a pattern (defined in Allegro Prolog[6]) to match a certain history state, a number of equivalent alternative verbal responses and the affective state update command. We have defined just under 20 rules, which make the agent respond to a variety of situations. The rules can be roughly divided into three categories: affectively-neutral rules, rules causing current affective valence to nudge slightly towards the positive end, and rules resulting in a move towards the negative end. The values of positive/negative affect changes are defined individually in each rule. When dominated by positive affect, the agent smiles; negative affect makes the agent appear sad. The following are examples of situations, with descriptions of corresponding responses and affective changes in the agent's appearance:

---

[5] http://www.cepstral.com/ — Cepstral Text-to-Speech engines.
[6] http://www.franz.com/products/prolog/ — Allegro Prolog – integrated extension for Common Lisp.

First-time login – agent introduces itself and welcomes the user with an extended message; the agent's affective state is changed so that the agent smiles.
Selection of a new agent – agent introduces itself and smiles.
Submission with a few errors – agent starts with a friendly introductory phrase and reads the errors; the agent's affective state is nudged slightly towards the negative side. If this attempt happens to come in a series of unsuccessful attempts, the agent will have an unhappy/concerned look as the result of its affective valence being dampened by the repeated triggering of this rule. However, if this is a single unsuccessful attempt, the change in the agent's appearance will be subtle.
Submission with a single error – the agent starts an encouraging phrase and presents the error; affective valence is nudged slightly to the positive direction.
Correct solution – the agent congratulates the user with solving the problem in one attempt and while happily smiling suggests that the user try another problem.

There are also rules defining behaviours for repeated submissions of identical solutions, repeated abandoning of problems, logging out etc. Precedence of session states is implicitly encoded in the order of the rules; during the rule matching process, the first match results in the corresponding action on behalf of the agent. Both male and female agents are guided by the same set of rules.

2. The Experiment

In July 2006, we recruited 20 volunteers (16 male and 4 female) from the third and fourth year students at the University of Canterbury. All participants were familiar with the EER model. The disparity between the users' levels of EER expertise was not an important factor, because the study was aimed at collection of qualitative data only. The experiment was carried out as a series of individual sessions. At the start of a session, a participant was provided with a verbal description of the task. Participants were expected to spend 45 to 60 minutes solving problems. The participants were asked to choose an agent before they were able to navigate to the workspace. During the session, the participants were free to choose a different agent at any time. At the end, the participants were required to fill out a questionnaire, for assessing their experience of learning with the agent and their perception of the agent's persona.

The first-choice preference was given to female agents (14 vs. 6) irrespective of the participants' sex. Among male participants only 38% chose a male agent, while all female participants chose a female agent at the start of the session. On the basis of questionnaire responses, it appears that male agents won in their effectiveness and overall impression on the participants. The six participants who chose a male agent reported they enjoyed EER-Tutor more, giving it a rating of 4.1 ($\sigma = 0.4$) out of 5, compared to the rating of 3.7 ($\sigma = 0.7$) by the participants who chose the female agent. The average learning success (rated on the scale of 1 to 5) reported by participants with male agents is higher then the female agents' group: 3.5 ($\sigma = 0.5$) vs. 3.1 ($\sigma = 0.7$). This learning estimate difference is consistent with the actual learning outcomes, measured by the number of constraints learned during the session: 3.3 ($\sigma = 4.8$) for male agents vs. 2.1 ($\sigma = 2.3$) for female agents. However, the small number of participants does not allow us to treat these results as being statistically reliable.

We attribute the apparent association between higher ratings and better learning outcomes for the male agents to the difference in the quality of male and female Cep-

stral TTS voices used. The male TTS voice sounds more human-like then the female voice; even though the participants were not specifically asked to rate the quality of TTS voices, 50% of the participants who used female agents stated that they were frustrated by or disliked that voice. No such comments were made about the male voices. Our interpretation of the voice quality effect is supported by the Cognitive Load Theory [20], which states that processing unnatural-sounding or machine-like voices imposes higher cognitive demands on learners than natural voices do.

The general response to the agents was positive – 75% rated the agents as a useful feature. At the same time, half of the participants who thought the agent's presence was unnecessary rated audible narration as useful. Overall, the participants were enthusiastic about narration – 50% stated that narration was the most helpful feature, because it made it possible for them to keep their eyes on the diagram and begin correcting errors while listening to the narration. Participants commented that this helped save time and enabled them solve problems faster.

Both male and female agents' responses and appearance were rated as adequate by around 90% of both male and female participants. Participants' feedback indicates that the agents' persona was appreciated for not being "in-your-face". Some users commented that they liked the "feeling of company" and "human presence" created by the agents. Some users made comments suggesting that the agents should be more active and versatile in their behaviour; others commented on being distracted by the agents' movements, such as blinking and breathing, in the background.

The study highlighted the need for the agents' behaviour to appear more natural. Many comments suggested enabling the user to have greater control over the agents' behaviour and voice properties. For example, some users wanted the agents to be more dynamic, even "have them fly over the workspace" and "make emotional changes more apparent", while others wanted the agents to be less obtrusive; some participants said they would like to be able to control the speed of the narration to make it faster or slower. Some users (around 25%) did not pay attention to the agents' affective appearance, but a small proportion (around 15%) commented on the agents "flipping out" and getting "too emotional" too soon. One participant commented that he or she felt "emotionally blackmailed and manipulated by the agent".

3. Conclusions and Future Work

We described an approach to modelling affective pedagogical agent persona and reported evaluation results. The persona was designed as a set of rules corresponding to the interaction states which require responses from a pedagogical, affective or social point of view. Our approach is characterised by simplicity, flexibility and scalability. The persona can be developed incrementally by adding new rules; the larger the number of rules, the more versatile and interactive the persona is. With this approach it is possible to develop different sets of rules defining different types of personas.

We created a mentor-like persona, mimicking a somewhat informal HHI interaction with passive emotional support. Four animated characters were designed aimed at soliciting perception of the agents' mentor persona. The results indicate a good outlook on the viability of the suggested approach and we will enhance our implementation in future work. Our agents' personae secured positive ratings during the evaluation; however, the lack of scale in the study and the confounding factor of difference between the quality of male and female TTS voices does not allow us to make conclusions about

participants' preferences on the agent's sex and its effect on learning. We have learned that the quality of agents' voice is critical for maintaining the agents' rapport with the users and maintaining the flow of the learning process.

This study also reveals the breadth of user preferences when it comes to interacting with affective pedagogical agents and suggests that finding an ideal persona to "click" with all users is unrealistic. Some learners see the agent as a distraction, while others wish the agent to be more interactive and entertaining. Needless to say, affective agents need to respect human individuality in the style of interaction. Even though the study shows the agents were a well-received addition to EER-Tutor, the participants' comments indicate the need for making the agents more flexible. Audible narration was welcomed by most participants irrespective of their expectations of the agents' persona.

In future work, we intend to extend our system to identify students' affective states via real-time facial feature tracking and physiological sensors. Incorporating data from the sensory channel and changing the agents' persona rules will give the agent a finer level of affective awareness.

References

[1] G. W. Allport, *Attitudes*, in C. A. Murchison, ed., *A handbook of Social Psychology*, 1935, pp. 798-794.
[2] A. L. Baylor, Y. Kim, *Pedagogical Agent Design: The Impact of Agent Realism, Gender, Ethnicity, and Instructional Role*, in J. C. Lester, R. M. Vicari and F. Paraguaçu, eds., *Intelligent Tutoring Systems*, 2004, pp. 592-603.
[3] T. W. Bickmore, *Unspoken Rules of Spoken Interaction*, CACM, 47, 2004, 38-44.
[4] C. Conati, H. Maclaren, *Evaluating a Probabilistic Model of Student Affect*, *Intelligent Tutoring Systems*, 2004, pp. 55-66.
[5] C. Gama, *Metacognition in Interactive Learning Environments: The Reflection Assistant Model*, *Intelligent Tutoring Systems*, 2004, pp. 668-677.
[6] D. Goleman, *Emotional Intelligence*, Bantam Books, New York, 1995.
[7] J. Gratch et al., *Creating Interactive Virtual Humans: Some Assembly Required*, IEEE Intelligent Systems, 17, 2002, 54-63.
[8] W. L. Johnson, J. W. Rickel, J. C. Lester, *Animated Pedagogical Agents: Face-to-Face Interaction in Interactive Learning Environments*, Artificial Intelligence in Education, 11, 2000, 47-78.
[9] J. Klein, Y. Moon, R. W. Picard, *This Computer Responds to User Frustration: Theory, Design, and Results*, Interacting with Computers, 14, 2002, 119-140.
[10] H. Klingzing, T. Tisher, *Expressive nonverbal behaviors; A review of research on training with consequent recommendations for teacher education*, in J. Raths and L. Katz, eds., *Advances in Teacher Education*, Ablex Publishing, 1986.
[11] J. Lester et al., *Deictic and emotive communication in animated pedagogical agents*, Embodied Conversational Agents, 2000, pp. 123-154.
[12] J. Lester, S. Towns, P. Fitzgerald, *Achieving Affective Impact: Visual Emotive Communication in Lifelike Pedagogical Agents*, Artificial Intelligence in Education, 10, 1999, 278-291.
[13] R. E. Mayer, *Multimedia Learning*, Cambridge University Press, 2001.
[14] P. Mishra, M. Nicholson, S. Wojcikiewicz, *Does My Word Processor Have a Personality? Topffer's Law and Educational Technology*, Adolescent and Adult Literacy, 44, 2001, 634-641.
[15] S. Ohlsson, *Learning From Performance Errors*, Psychological Review, 103, 1996, 241-262.
[16] A. Ortony, G. Clore, A. Collins, *The Cognitive Structure of Emotions*, Cambridge University Press, 1988.
[17] J. Preece et al., *Human-Computer Interaction*, Addison-Wesley, New York, 1994.
[18] B. Reeves, C. Nass, *The Media Equation.* Cambridge Publications, 1996.
[19] P. Suraweera, A. Mitrovic, *An Intelligent Tutoring System for Entity Relationship Modelling*, Artificial Intelligence in Education, 14, 2004, 375-417.
[20] J. Sweller, J. van Merrienboer, F. Paas, *Cognitive Architecture and Instructional Design*, Educational Psychology Review, 10, 1998, 251-296.
[21] K. Zakharov, A. Mitrovic, S. Ohlsson, *Feedback Micro-engineering in EER-Tutor*, in J. Breuker, ed., *Artificial Intelligence in Education*, IOS Press, 2005, pp. 718-725.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Serious Use of a Serious Game for Language Learning

W. Lewis JOHNSON
Alelo, Inc. and the University of Southern California
ljohnson@alelo.com

Abstract. The Tactical Language and Culture Training System (TLCTS) helps learners acquire basic communicative skills in foreign languages and cultures. Learners acquire communication skills through a combination of interactive lessons and serious games. Artificial intelligence plays multiple roles in this learning environment: to process the learner's speech, to interpret and evaluate learner actions, to control the response of non-player characters, to generate hints, and to assess the trainee's mastery of the skills. AI is also used to assist in the authoring process to assist in the generation and validation of lesson content. This paper gives an overview of the current system, and describes the experience to date in transitioning the system from research prototype into a training system that is in regular use by thousands of users in the United States and elsewhere.

Keywords. Game-based learning, pedagogical agents, second language learning

Introduction

The Tactical Language and Culture Training System (TLCTS) is designed to help learners quickly acquire basic communication skills in foreign languages and cultures. Learners acquire knowledge of foreign language and culture through a combination of interactive lessons and interactive games that give trainees concrete contexts in which to develop and apply their skills. It focuses on spoken communication, nonverbal communication, and cultural knowledge relevant to face-to-face communication. TLCTS is an example of a serious game applied to learning [11]: it utilizes game design techniques to promote learning, e.g., by providing learners with missions to achieve, supporting fluid gameplay in the form of simulated conversations with non-player characters, and continual feedback on learner performance within a game scenario context. It utilizes artificial intelligence to engage in speech recognition and dialog with artificially intelligent characters, and to estimate learner mastery of target skills. It also employs artificial intelligence to assist in the creation and validation of instructional content.

This paper provides an overview of the system and its architecture. In then focuses on the process of transitioning TLCTS from a research prototype into a robust learning tool in wide use in the US military and elsewhere.

1. System Overview

Each TLCTS training course includes the following major components. The Skill Builder consists of interactive lessons focusing on task-relevant communication skills. The Arcade Game and Mission Game are interactive games that give trainees opportunities to develop and practice communication skills. The Web Wizard provides reference material, including glossaries and explanations of the grammatical structure of the phrases used in the lesson material.

Figure 1 shows example screens from the Tactical Iraqi course, designed to help people learn Iraqi Arabic language and culture. The image on the left shows a cultural notes page from the Skill Builder, which illustrates a common greeting gesture in the Muslim world, the palm-over-heart gesture. The image on the right shows the learner's character in the Mission Game greeting an Iraqi non-player character with that gesture.

The Skill Builder and the game experiences are both speech-recognition enabled. In the Skill Builder learners practice vocabulary and phrases, and complete exercises and quiz items that require speaking and understanding spoken language. In the Arcade Game, the learner gives spoken commands in the target foreign language to direct his or her character to move about a game world. In Mission Game, the learner speaks on behalf of his character. This is taking place in the screenshot on the right side of Figure 1, while the character performs a hand gesture that the learner had previously selected from a menu.

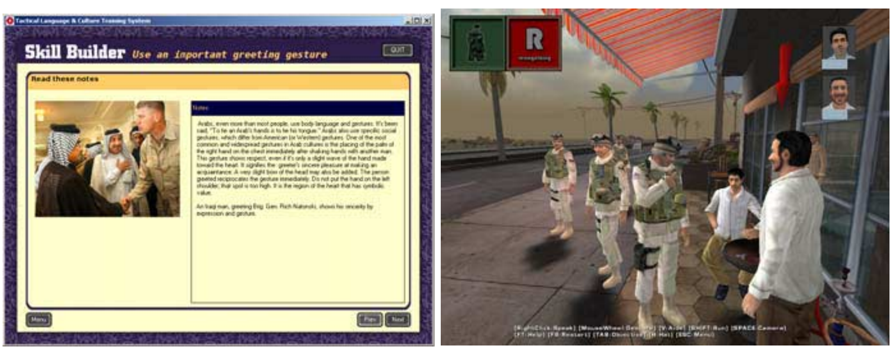


Figure 1. Screenshots from the Tactical Iraqi Skill Builder and Mission Game

The lessons and exercises in the Skill Builder progressively prepare the learner for employing their communication skills in the free-play Mission Game, and ultimately in the real world. Figure 2 shows an intermediate point in this progression, a so-called active dialog. Here the player character (at left) is engaged in a conversation with the head of the household (at right, under the red arrow), in the context of searching a house for weapons and contraband. The screen at bottom shows the most recent phrase that the speech recognizer detected, iftaH il-baab (open the door), and a hint for the next operation to perform (to tell the head of household to put the women and children in a separate room). In active dialogs the learners are guided through the dialog by means of these hints, whereas in the Mission Game the learner does not receive hints unless he or she specifically requests them.

All TLCTS content is highly task-based [4], i.e., instruction focuses on what learners need to know in order to accomplish particular tasks. The task-based approach is very appropriate for game-based learning environments such as this. The task-based approach carries over to the Skill Builder lessons as well, since lessons are focused on acquiring

particular skills relevant to particular types of situations. The content development method used for TLCTS content explicitly takes this into account. Each Skill Builder lesson is annotated as to the particular skills that it emphasizes, as are the Mission Game scenes. This helps authors to ensure that the Skill Builder adequately prepares learners for the Mission Game scenes. The scenes and lessons are automatically cross-indexed in terms of skills, making it possible for trainees to focus in on the particular lessons and lesson pages that they need to study in order to complete the game scenes successfully.

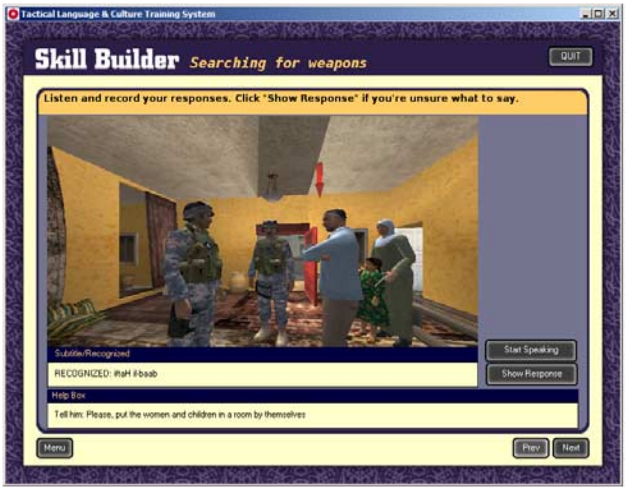


Figure 2. An active dialog in Tactical Iraqi

As the learner works with the software, the software automatically tracks each instance when the learner applies a skill, and uses it as probabilistic evidence of mastery of the skill, akin to knowledge tracing. As Beck and Sison [1] have noted such evidence is inherently uncertain in speech-enabled application where there is a possibility of recognition error, however that can easily be incorporated into a knowledge tracing model by treating recognition errors as just another source of “guesses” (false positives) and “slips” (false negatives). In any case, learners see that the mastery estimates progressively increase with practice, which motivates them to keep practicing until they reach high mastery scores.

While other language learning systems employ speech recognition technology, and support simulated dialogs [2, 3, 6, 7, 8], and employ AIED technology [5], TLCTS is unique in the extent to which it employs these technologies to support the acquisition of face-to-face communication skills. It has been used to develop complete learning environments covering many hours of training.

2. Architecture

The following is a brief summary of the architecture of TLCTS. More detailed descriptions of earlier versions of the architecture are available in other publications [10, 12, 13].

The initial version of the TLCTS prototype was developed using a combination of software tools. The first version of the Skill Builder was authored in ToolBook, and the

Mission Game was implemented as an end-user modification ("mod") of the Unreal Tournament 2003 PC video game. The Web Wizard was implemented in Dynamic HTML and viewed through a Web Browser. This mixed approach was essential in order to develop prototypes quickly that could be presented to stakeholders, however it proved to be problematic for users. Formative evaluations of TLCTS [9] indicated that this mixed implementation was cumbersome for users, and discouraged them from shifting between components of the learning environment. A new version of the Skill Builder was therefore implemented using Unreal Tournament user interface objects. This transition as accomplished in the following stages. First, an XML specification language was developed for Skill Builder lesson content, and the ToolBook implementation was modified to generate lesson pages from those XML descriptions. Then, a new display generator was created in Unreal that constructs display pages for each page in the XML description. In some cases the same XML description is used to generate multiple display pages: the author specifies a lesson page, and then the page generator automatically generates exercise pages that test the learner's recall of the items on the lesson page.

Once the Skill Builder was integrated into Unreal, learners were more inclined to switch between components of TLCTS as needed. Further steps have since been taken to integrate and simplify the TLCTS architecture, as is described below.

3. Transition into Serious Use

A complete prototype of Tactical Iraqi was completed in June 2005. At this point a new phase of development and evaluation began, leading ultimately to transition into regular use.

Achieving transition of Tactical Iraqi would require overcoming a number of technical and nontechnical obstacles. Tactical Iraqi was developed under sponsorship of a research agency (DARPA). No military service had requested it, or had made plans to acquire it. In fact because it was a research prototype, including new and untested technologies, there was little reason to believe that it would be suitable for regular military use. US military already had access to a range of language learning materials, including an Army-wide license for Rosetta Stone. TLCTS required up-to-date videogame computers to run, which most military units do not have for training purposes. The US military places imposes severe security restrictions on software that runs on its networks, so military units that were interested in using Tactical Iraqi would have to purchase entirely new sets of computers that were not linked to the military network. Finally, military units undergoing training have highly packed training schedules; many military commanders felt that their troops did not have the time to commit to a training program such as Tactical Iraqi, or felt that language and culture training was less important than other types of training.

To start this process, DARPA funded an initial evaluation study with the Marine Corps. A series of two pilot two-week training courses were administered at a Marine Corps training facility at Camp Pendleton, CA, on new laptop computers provided expressly for this purpose. Each training course showed promising results, and also identified problems that were collected in subsequent versions of the software and programs of instruction. For example, in the first training course trainees discovered that they could play the Unreal Tournament 2003 game without Tactical Iraqi, and were tempted to do so; therefore we further modified Unreal so that it could not be run separately. The second pilot evaluation started showing significant promise, while still pointing to areas where further improvements are required. Twenty Marines participated

in this study. All had been directed by their command to participate in the study (in Marine jargon, they were "voluntold" to participate). The participants were all enlisted personnel, and only one had any appreciable knowledge of Arabic at the beginning of the course. Of the twenty participants, nine had previously been deployed to Iraq. Overall, the strongest predictor of success with Tactical Iraqi proved to be whether or not they had previously been to Iraq, and therefore understood the importance of the training provided by Tactical Iraqi. Among the participants who had previously been to Iraq, 78% felt at the end of 50 hours of training that they had acquired a functional ability in Arabic within the scope of the missions being trained. Among this group, the trainees all gave the software a subjective evaluation of at least 4 on a scale of 0 to 5, and the participants who gave it a 4 generally had a number of constructive suggestions about how to make the software better. Among the participants who had not previously been to Iraq the percentage that believed that they had obtained functional ability was much less (22%), and the average subjective evaluation was somewhat lower, 3.73 out of a possible five. One reason why this group gave the software a lower score was that the initial prototype was focused on the requirements of Army personnel. One participant reported that because the software did not teach how to say "I am a US Marine" in Arabic, the software was worthless, and he gave it a rating of 0 out of 5.

The training course and delivery platform was subsequently improved based on feedback from these evaluations and other military beta testers. The platform was made to be configurable, so that users can choose whether to operate the system in Army, Marine Corps, or civilian mode. The choice of user classification determines the dress of the characters in the games in the Skill Builder dialogs, as well as the choice of content in the Skill Builder and the choice of dialog in the Mission Game. Customizations for each class of user were necessary in part because soldiers and Marines have different enlisted military ranks, and therefore use different forms of address from each other and from civilians. To support these customizations, the XML specifications for Skill Builder content were augmented so that individual lesson pages can be annotated as being appropriate for particular classes of users.

Although it was convenient to develop the initial prototype of the Mission Game as a mod, this proved to be a barrier to transition. Users did not want to have to install and run multiple programs in order to run TLCTS. It therefore became necessary ultimately to acquire a license to the Unreal Engine game engine underlying Unreal Tournament, and integrate all capabilities into a single executable package. It was also necessary to rework the user interface to include screens that are more readable and conducive to learning, as shown in the figures shown above. The completed learning environment retains very little of the look and feel of the original game environment, except for some of the keyboard controls used to navigate through the simulated game world.

Software testing is particularly critical for learning environment such as TLCTS, which includes complex AI-based interactive characters, speech recognition, learner modeling, and other advanced capabilities. We therefore had to develop a series of analysis tools and test harnesses to validate the learning content. In some cases these can be applied at authoring time, e.g., to check content for misspellings or words that have not yet been introduced. Some testing is performed when the first executable prototypes of the training systems are created, e.g., to test the dialog models in the non-player characters. A testing interface was created that displays turn by turn the possible dialog moves that the non-player character is expecting that the learner might say next, and then shows the character's response. Such testing frequently reveals possible dialog moves that the authors failed to anticipate, or inappropriate actions taken by the non-player characters.

Evaluation of the speech recognizer was particularly important and challenging. One reason for this is that standard measures of speech recognition accuracy (e.g., word error rate) are not very relevant to speech-enabled learning environments such as TLCTS. Speech recognition performance needs to vary based on the context in recognition is performed, and the skills of the learner. In the Mission Game speech recognition is used to recognize the learner's intent (i.e., category of communicative act), whereas the Skill Builder lessons tend to focus more on detecting and correcting common language errors. If a learner is working on an advanced lesson but his or her pronunciation is poor we actually prefer the word error rate to be high, so that learners will be motivated to improve their pronunciation. We therefore collected data sets (recorded by TCLTS) of users using the system in different contexts, and used these both to evaluate and retrain the performance of the speech recognizer. This enabled us to improve speech recognition performance so that performs acceptably well as needed, with high reliability.

The learning environment itself is just one component of the set of training materials that needed to be developed. User manuals and staff development seminars (known as "train-the-trainer" sessions in the military) needed to be developed. Although educators are familiar with the importance of staff training, such notions are not common in the videogame community. As a consequence many prospective users supposed that a software CD is all that they needed to start training. We have continued to add tutorials and background materials to make the training systems more self-explanatory, and we expect that users can gain training benefit without guidance or orientation. However in a learning environment as complex as TLCTS it is common for learners to overlook important features of the system, or use the system in sub-optimal ways.

Programs of instruction (i.e., curricula) needed to be developed, that provided trainers with guidance as to how trainees should make most effective use of the learning environments. These were developed mainly through collaborations between TLCTS project staff (mainly this author) and early adopters of the learning environment. Early adopters would sometimes develop their programs of instruction, of varying quality. This author then took examples of the better programs of instruction, generalized and extended them and incorporated them into a trainer guide which is made available to all users. The programs of instruction recommend that learners alternate between Skill Builder study and practice in the game components as they progress through the course. Although this might seem obvious, may learners and trainers fail to recognize the importance of this approach.

Summative evaluation of learning outcomes will also be an enabler of transition of TLCTS into use. However such evaluations are feasible only once a commitment has been made to make use of TCLTS in training. In the case of Tactical Iraqi, some military training centers, such as the Expeditionary Warfare School and the Battle Simulation Center at 29 Palms, CA, chose to become early evaluators and trial users of the learning environment. 29 Palms became a particularly important site – it has a laboratory of fifty computers installed with Tactical Iraqi, used both for training and demonstration. Marine units started using the Center in an organized fashion for training starting in the summer of 2006. We have recently collected log data from several hundred Marines who have used the Center, and are currently analyzing their learning gains and performance.

4. Status and Future Work

Up to the present time Alelo and USC have distributed over 6000 copies of Tactical Iraqi and Tactical Pashto. The total number of copies in use is likely to be substantially greater

than that, since a number of training centers copy the software and distribute copies to their users. A number of US military posts and training have set up computer labs for training, in the United States, Europe, and in Iraq. Some of these centers have made a heavy commitment to using the software. For example, the 3rd Infantry Division trained 2,000 soldiers with Tactical Iraqi prior to their next deployment to Iraq. The Marine Corps Expeditionary Warfare School (EWS) has integrated Tactical Iraqi as a required part of its curriculum, and has made the software available on computers throughout the School. Students also receive copies of the software which they can take home with them.

Use of Tactical Iraqi continues to expand. This is happening because learners stakeholders at all levels evaluate it positively and advocate its use. Learners at all ranks who have worked with it are convinced that it is effective in acquiring relevant communication skills. Instructors such as those EWS are convinced of its value, and also report that the current version is relatively free of bugs and technical glitches. Commanding officers have become convinced of the value of this training, and in more than one case general officers have taken it upon themselves to train with Tactical Iraqi, in part so as to set an example for their troops.

Military servicemembers with .mil accounts are authorized to download free copies of Tactical Iraqi; over a hundred copies are downloaded each month, and this number steadily increases. The Marine Corps plans to set up additional training labs for use with Tactical Iraqi. As soon as Tactical Iraqi is approved for use on military networks, it will distributed across those networks as well.

Meanwhile Tactical Pashto is starting to be used. Extensive use by the 3rd Marine Expeditionary Force in Okinawa and the Marine Corps Mountain Warfare Training Center in Bridgeport, CA in June, 2007.

Additional courses are currently under development. One course, Tactical French, is designed to prepare military personnel to train local military forces in Sahel Africa, with Chad at the notional focus. Another prototype course, Mission to France, is designed for businessmen to help them conduct business in France; this course takes place a slightly fictionalized version of Nice, France. This is the first course developed without a military focus. However, we anticipate that the same task-based approach can be used effectively in this non-military application as well. We are engaged in a collaborative project with McKinley Technology High School in Washington DC to develop the Tactical French. The students in this school are interested in learning about Francophone Africa, and are also interested in acquiring videogame development skills. We therefore plan to engage these high school students initially to test early versions of the training system, and then contribute to the development of the 3D virtual world in the game.

Meanwhile, content development for Tactical Iraqi continues. New lessons and scenarios are being created for new missions of current relevance in Iraq, such as the vocabulary needed to train Iraqi defense forces and conduct patrols. Plans are underway to extend the Tactical Iraqi curriculum up to the point where trainees who complete the course will be able to pass a basic spoken proficiency test in Iraqi Arabic, which will entitles Marines to pay bonuses.

Through this and other steps we anticipate that Tactical Iraqi will become more tightly integrated into the practice of language and culture training in the US military. Military forces in other countries are also interested in Tactical Iraqi and Tactical Pashto, and so we plan to make these courses available to military forces in other countries in the near future.

Research continues to be conducted in a number of areas. A new multilingual content authoring tool named Kona is currently under development, which allows authors specify the content of lesson pages. Specifications may include the source language (e.g., English),

the target written language (e.g., written Arabic), phonetic transcriptions used for speech recognition, and pronunciation glosses designed to aid the learner. We automate the generation of some of these transcriptions, e.g., so that pronunciation glosses are generated automatically following rules specified by the instructor or content author. Prototype Skill Builder implementations have been developed for handheld game players and Pocket PCs, these could provide valuable just-in-time reinforcement for skills acquired in the learning environment. Meanwhile work continues on improving the provision of hints (companion submission to this conference), tracking learner activities in the learning environment against training plans, providing added capabilities for detecting pronunciation errors and providing feedback, and assisting the authoring process.

Acknowledgments

The author wishes to acknowledge the members of the development teams at Alelo and USC for their contributions to the work described here. The author also wishes to acknowledge the assistance of the early adopters of TLCTS products throughout the military services. This work was sponsored by DARPA, USSOCOM, and USMC PM TRASYS. It opinions expressed in the paper are those of the authors' and do not reflect those of the sponsors or the US Government.

References

[1] J. Beck, & J. Sison, Using knowledge tracing to measure student reading proficiencies. Proceedings of the 7th International Conference on Intelligent Tutoring Systems, Springer-Verlag, 2004.

[2] J. Bernstein, A. Najmi, & F. Ehsani, F., Subarashii: Encounters in Japanese Spoken Language Education. CALICO Journal **16** (3) (1999), 361-384.

[3] W.J. DeSmedt, Herr Kommissar: An ICALL conversation simulator for intermediate German. In V.M. Holland, J.D. Kaplan, & M.R. Sams (Eds.), *Intelligent language tutors: Theory shaping technology*, 153-174. Lawrence Erlbaum, Mahwah, NJ, 1995.

[4] C.J. Doughty & M.J. Long, Optimal psycholinguistic environments for distance foreign language learning. Language Learning & Technology **7**(3), (2003), 50-80.

[5] G. Gamper, G. & Knapp, J.: A review of CALL systems in foreign language instruction. In J.D. Moore et al. (Eds.), *Artificial Intelligence in Education*, 377-388. IOS Press, Amsterdam, 2001.

[6] H. Hamberger, Tutorial tools for language learning by two-medium dialogue. In V.M. Holland, J.D. Kaplan, & M.R. Sams (Eds.), *Intelligent language tutors: Theory shaping technology*, 183-199. Lawrence Erlbaum, Mahwah, NJ, 1995.

[7] W.G. Harless, M.A. Zier, and R.C. Duncan, Virtual Dialogues with Native Speakers: The Evaluation of an Interactive Multimedia Method. CALICO Journal **16** (3) (1999) 313-337.

[8] V.M. Holland, J.D. Kaplan, & M.A. Sabol, Preliminary Tests of Language Learning in a Speech-Interactive Graphics Microworld. CALICO Journal **16** (3) (1999) 339-359.

[9] W.L. Johnson, W.L. & C. Beal, Iterative evaluation of an intelligent game for language learning, AIED 2005, OS Press, Amsterdam, 2005.

[10] W.L. Johnson, C. Beal, A. Fowles-Winkler, U. Lauper, S. Marsella, S. Narayanan, D. Papachristou, and Vilhjalmsson, H., Tactical Language Training System: An Interim Report. ITS 2004. Berlin: Springer-Verlag, 2004.

[11] W.L. Johnson, H. Vilhjalmsson, & S. Marsella, Serious games for language learning : How much game, how much AI ? AIED 2005. IOS, Amsterdam, 2005.

[12] W.L. Johnson, S. Marsella, & H. Vilhjálmsson, H. The DARWARS Tactical Language Training System. Proceedings of I/ITSEC 2004.

[13] H. Vilhjalmsson, & P. Samtani, P., MissionEngine : Multi-system integration using Python in the Tactical Language Project. PyCon 2005, 2005.

Representation

This page intentionally left blank

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Helping Courseware Authors to Build Ontologies: The Case of TM4L

Darina DICHEVA and Christo DICHEV
Winston-Salem State University,
601 M.L.K. Jr. Dr., Winston Salem, N.C. 27110, USA

Abstract. The authors of topic map-based learning resources face major difficulties in constructing the underlying ontologies. In this paper we propose two approaches to address this problem. The first one is aimed at automatic construction of a "draft" topic map for the authors to start with. It is based on a set of heuristics for extracting semantic information from HTML documents and transforming it into a topic map format. The second one is aimed at providing help to authors during the topic map creating process by mining the Wikipedia knowledge base. It suggests "standard" names for the new topics (paired with URIs), along with lists of related topics in the considered domain. The proposed approaches are implemented in the educational topic maps editor TM4L.

Introduction

The web is changing dramatically the way teachers teach and students learn. For almost any topic now there are online resources that are more current and diverse in perspective than the printed media. For example, computer science master students would find about 70% of their reading on the web. Despite the abundance of online information though, many students fail to get the best out of it: even Google's smart keyword search provides hundreds of hits to browse. In addition, the web information is not always accurate, credible or reliable. For e-learning this implies a need of repositories, complimentary to web, with trusted information, organized appropriately for easy access by learners. Ontology-driven learning repositories, supported by intuitive interface and efficient navigation and search capability for easy exploration, come as a promising solution.

Ontology-driven learning repositories need authoring tools that support both the ontology creation and semantic annotation of resources. Currently available general purpose ontology languages and tools though tend to impose high entrance barriers for end users. They typically assume that authors are able to conceptualize domains of interest in terms of concepts, concept types, and relationships between concepts. However, courseware authors are largely used to organizing learning content in terms of modules, chapters and subchapters (i.e. of "concept containers"), instead of concepts [1]. The task of providing a complete and systematic list of all concepts that should be covered by a learning unit is not trivial. Additional "ontological challenges" include using widely agreed names for the concepts and inter-relating them with meaningful relationships. So, authors do need assistance in conceptualizing subject domains.

Topic Maps for e-Learning (TM4L) [2] is an environment for creation and use of topic-focused, ontology-driven learning repositories, based on the Semantic Web

technology Topic Maps (TM) [3]. In the Topic Map paradigm, an ontology is an accurate description of the essential entities and relations in the modeled domain and can be represented as a set of topics linked by associations.

Compared to the manual topic map authoring offered by TM4L, an automatic extraction of topical structures seems very appealing. However, at present there is no technology available to provide automatic topic, relationship, and resource extraction in conceptually clear, intuitive, accurate, and scalable fashion. Indeed, fully automatic topic map construction is not feasible yet, especially in the education domain, where many subject and resource specific decisions must be made to adequately specify a learning collection. Thus our goal was to extend TM4L functionality, so as to support authors with limited knowledge of IT and ontologies as much as possible.

The proposed support consists of automatic extraction of topic map constructs (concepts and relationships) from specified web pages and using them to build a "draft" topic map for the author to start with. We have analyzed and compared different text extraction approaches (including machine learning and natural language processing) and have chosen to extract topics and relationships by crawling a website and using the semantic information that can be extracted from the HTML markup. Our idea was inspired from a study on the feasibility of using three HTML tags as proxies for semantic content: link text, heading, and comment [4]. In that study, human experts have analyzed a large number of web pages and have found that 61% of the link text was "helpful" or "very helpful" in indicating semantic content. This seemed quite encouraging since the web pages have been arbitrary chosen.

We designed and implemented a plug-in for TM4L including two options for automatic extraction of topics and relationships: from a website specified by the author and from the Wikipedia and Wikibooks websites.

1. Topic Map Object Extraction from a Website

This aspect of our support for TM authors was motivated by the fact that there is a significant amount of semi-structured information on the web. HTML documents, for instance, are structured for rendering purposes but their structure can also be used for extracting some semantic information. For example, list items can be transformed into members of "whole-part" relationships; items marked up as "bold" or "italic" can be considered semantically important, etc. Thus our goal was to find out what semantic information can be extracted from the HTML markup of web pages and included in the intended 'draft' topic map. Since there are no formal rules for extracting semantic information, heuristic approaches are practically feasible. From this viewpoint we can set some heuristics for HTML tables, such as "A row represents a topic, a column represents an occurrence". Such observations, combined with experiments with various types of information in HTML format led us to propose the following *heuristic rules*.

1.1. Defining 'Page' Topics

A 'draft' topic map consists of topics and relationships. These objects are extracted by crawling a specified web site. We differentiate between two types of topics: topics that reflect the web site topology and topics extracted from the text of the web pages.

Rule 1: For each web page visited by the crawler, a new topic is created in the topic map. We call these topics "page" topics.

Rule 2: All the topics created in the process of parsing a specific web page are sub-topics of the "page" topic for that page.

Rule 3: Naming of "page" topics: The theme of interest, provided by the user, is used as a name of the "page" topic, corresponding to the entry page for crawling the site; all other "page" topics are named using the text in the corresponding "anchors".

1.2. Information Extraction from Heading Tags, List Element Tags, and Table Tags

Rule 4: Heading represents a topic that is more general that the topics extracted from the text below it (if any).

Rule 5: The topics extracted from headings of different levels can be organized in a taxonomy reflecting headings' level (1, 2, etc.).

Rule 6: Heading tags on a referenced (through an 'anchor' element) web page are considered as structurally related to the "page" topic of the referencing page. Thus the result of parsing heading tags will consists of a set of topics named by the text enclosed in the heading elements and connected to the "page" topic of the referencing page.

Rule 7: The topics extracted from list item tags are more detailed (sub-topics) of the topics contained in the list tags. The list-list item relationship between any two topics is modeled by a "child-parent"-type relationship. Table of contents expressed as a bulleted list in HTML has an isomorphic image in terms of a "whole-part" tree in TM.

Rule 8: The topics extracted from the cells of a table column are related, since they can be considered as values of the same attribute (represented by the column header).

Rule 9: The topics extracted from the cells of one column in a table are subtopics of the topic corresponding to the column header.

The difficulties in extracting relations from text are largely recognized, but we can try capturing at least the relevancy of topics that appear in the same HTML elements:

Rule 10: Group the topics extracted from the same HTML element together, since this grouping indicates some kind of relatedness of the topics.

2. Topic Map Object Extraction from the Wikipedia and Wikibooks Websites

Since names are the primary mechanism for denoting subjects in human/machine discourse, we need sharable topic names. However, deciding, from one side, what concepts (topics) to include in an ontology and, from another, which are the widely agreed names for the selected concepts is a big challenge for the topic maps authors. We address this dual challenge by proposing the use of Wikipedia and Wikibooks as information sources in the automatic creation of draft topic maps. Manual topic extraction is not feasible since it is not easy from a given Wikipedia article to locate all topics related to it.

2.1. Why Wikipedia?

The underlying structure of a concept-based learning repository usually can not be derived from a single textbook or course syllabus. Besides the fact that textbooks and courses are frequently changed, their structures are often inconsistent. In addition, their titles and section names are generally arbitrary. If we aim at a reusable model, it should be founded on a more stable structure [1]. We can increase the level of consensus on a domain description by using Wikipedia as a provider of concepts (a vocabulary) for

defining topics or expressing queries (see Fig 1). So, the idea is to use the Wikipedia/Wikibooks corpus as a rich source of common sense conceptualization. This comes also in support of the consideration that ontologies are not just formal representations of a domain, but much more community contracts [5] about such formal representations. Wikipedia is popular as a reference and its concepts can be expected to have commitment by a wide audience. The authors of [5] have shown that the URIs of Wikipedia entries are surprisingly reliable identifiers for ontology concepts. The English version of Wikipedia contains more than 850,000 entries, which means it holds unique identifiers for 850,000 concepts. Because many people are contributing to Wikipedia, we believe that the terminology and naming convention there are more "standard" (agreed upon) than in other sources. Thus, Wikipedia can be used as a source of reliable and sharable concept names.

2.2. How to use it?

The TM4L extraction tool uses Wikipedia in the following way:

- As a source for proposing concepts relevant to a particular subject.
- As a source of "standard" concept (topic) names.
- As a source of URIs to be used as Public Subject Identifiers (PSI) [3] for concepts.

The duality of our Wikipedia use as a source of topics and identifiers reflects the human/computer dichotomy: topics are for the benefit of human users; PSIs are for the machines that need to ascertain whether two pieces of information are about the same subject. Technically, for a given sequence of keywords, e.g., "Operating Systems", a topic is selected from Wikipedia and Wikibooks that is the best match to the intended concept along with a set of related to it topics (see Fig. 1). Both sites are searched separately and the resulting topic collections are merged. In general, Wikibooks provides a better "table of contents" than Wikipedia (more useful for conceptual structuring of a course) however some topics are not covered there, since its content is more limited. The combined search in both sites provides a better coverage.

3. Implementation and Evaluation

3.1. Implementation issues

The TM4L extraction plug-in is written in Java and consists of Extraction Engine, Topic Map Builder, and User Interface. It uses WebSphinx and Java API for XML Processing (JAXP) that supports Document Object Model (DOM) API to process XHTML documents.

The heuristic rules proposed in Section 1 are used for parsing a specified unknown website. For choosing the Wikipedia/Wikibooks web pages to be processed, we use the Wikipedia search engine that returns Wikipedia links ordered by their relevance to the specified topic. The problem of using directly the standard URLs of Wikipedia articles is that an article corresponding to a required topic may not exits or a different type of addressing may be used. Our solution helps also in the case of misspelled words: the search engine returns the links in order corresponding to their relevance score and the highest ranked is taken for processing. For the Wikipedia/Wikibooks article search we

identified mark-up patterns that are applied consistently across all articles. For example, the majority of articles provide a table of content (TOC) to support better page navigation. Although, Wikipedia and Wikibooks apply a variety of page structuring patterns, we identified rules defining a TOC. When the algorithm cannot identify a TOC, it performs a search for section titles and links.

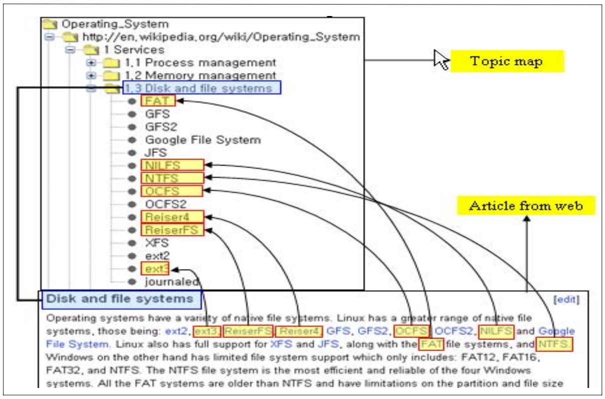


Fig 1. Topic hierarchy extraction from Wikipedia

The output of the extractor in both cases (from a specified unknown web page or Wikipedia) is a hierarchical structure of topics (Fig. 2). In Wikipedia, the level of nesting for a particular topic corresponds to its location in the page section-subsection structure. For example, “FAT” was found in “Disk and file systems”, which in turn is located in “Services” (see Fig. 1). The user can then select some of the proposed topics to be merged with his or her topic map.

3.2. Measuring the Effectiveness of the Extraction

To test our topic extraction algorithm we applied it to 35 web pages with different patterns of topic groupings. To illustrate the evaluation process we present here the evaluation results of applying the algorithm to two representative web pages: a topic specific page - Wikipedia Prolog (http://en.wikipedia.org/wiki/Prolog) and a general information page of the Math Forum Internet Mathematics Library (http://mathforum.org/library/toc.html). The assessment of the performance of the crawler and the proposed rules are based on the standard measures from Information Retrieval, *Recall* and *Precision*. For our task recall is interpreted as the number of relevant topics returned in a particular grouping, divided by the total number of relevant topics in that grouping. Precision is interpreted as the returned relevant topics, divided by the total number of topics returned by the crawler using the proposed heuristics. The evaluation results in terms of recall and precision are shown in Table 1 in Table 2.

A similar evaluation was performed for the information extraction from Wikipedia. Five instructors were asked to create topic maps with TM4L, each with 10 topics, based on their course syllabus. For each topic map, they had to report the number of all topics returned from Wikipedia, the number of relevant topics among them, and the total number of relevant topics in Wikipedia. A topic was considered relevant if the corresponding concept was covered in the course. The precision and recall values were then computed. (The tables are not presented here due to lack of space.)

Table 1. Recall values for the two websites

Web page	HTML Element	# Relevant Topics Returned	Total # Relevant Topics	Recall Value
Wikipedia/Prolog	Lists	50	50	1
	Tables	12	12	1
	Anchors	24	25	0.96
Math Forum	Lists	134	134	1
	Tables	10	22	0.45
	Anchors	134	156	0.86

Table 2. Precision values for the two websites

Web page	# Returned Relevant Topics	Total # Returned Topics	Precision Value
Wikipedia/Prolog	1055	1863	0.57
Math Forum	278	380	0.73

3.3. Discussion

Our preliminary assessments demonstrated an acceptable accuracy for the rules. Interpreting the topic extracted from an anchor element as the root for all topics extracted from the referenced page, generally resulted in a natural hierarchical structuring of the topics in the extracted tree and thus in a reasonable level of precision and recall. The text extracted from the anchor elements in most cases was appropriate for naming the "page topics". Heading tags allowed the extraction of topics related indeed by a "child-parent" relationship to the root topic. In a similar way, concepts extracted from headings were linked by a taxonomic relation to the concepts extracted from the subordinate sections. Topic extraction from HTML lists produced the best results in terms of precision and recall. The hierarchical topic structure captured from lists (see Fig. 2) reflects the nested nature of the corresponding list stricture thus preserving the intended relationship between the list items in the document.

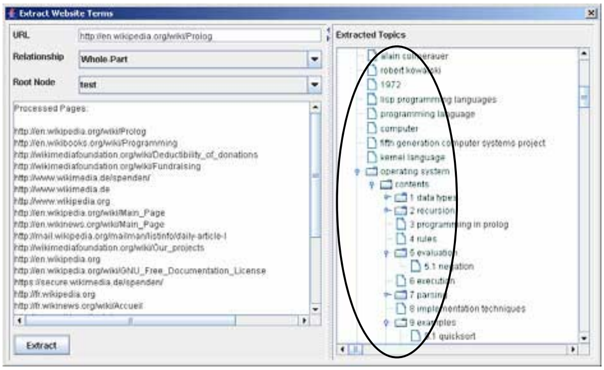


Figure 2. Results screen with list tags extraction

The evaluation results are encouraging since they were obtained without considering the impact of the user factor. Since the user selects the web site for

crawling, it is reasonable to expect that he or she will choose pages containing some sort of classification scheme. In such cases the recall and precision of the extracted topic structures will be comparable (if not better) to our results. We found out that our recall and precision results were not significantly different from the manually processed results reported in [4]. This is in favor of our algorithm since it extracts automatically a larger set of concepts while preserving the same rate of relevance. The precision value could be improved further by suggesting more heuristics.

4. Related Work and Conclusion

A vast number of methods have been proposed for text extraction, including text mining and NLP approaches. The novelty of the unstructured or semi-structured and highly noisy hypertext data, however, in many cases diminishes the efficiency of these classical methods. Among the most used are the clustering (unsupervised learning) and classification (supervised learning) methods. A recognized complication with clustering in the hypertext domain is the lack of agreement about what the different people expect a clustering algorithm should output for a given data set. This is partly a result of the difficulty to guess what their implicitly used similarity measures are, due to the typically very large number of attributes. The classic supervised learning methods (such as [6]) are not suitable for our goal, since they require initial training with a corpus of documents that are previously labeled with topics. For the same reason, the classical NLP approaches are also unsuitable. Language analysis should be specialized in a specific topical area. For example, [7] describes an NLP approach for extracting text to build a table of contents of pages on the web for specific concepts. The approach searches for definitions and subtopics of a given topic and preprocesses the extracted data using predefined verbs.

The OntoLT approach [8], used in a plug-in for the ontology editor Protégé, proposes automatic extraction of concepts and relations from annotated text collections. The annotation allows for automatic extraction of linguistic entities according to predefined conditions and using them to construct a shallow ontology of domain concepts and relations. This approach is not suitable for us, since it requires preliminary annotation of the text collection. Fortuna et al. describe a system for a semi-automatic ontology extraction from a document collection, using text mining algorithms including Latent Semantic Indexing (LSI), K-Means clustering, and keyword extraction to build their tool, which suggests to the user possible new topics [9]. When the user selects a topic (defined as a set of related documents), the system suggests subtopics of that topic, by applying LSI to the documents from the selected topic. While their goal of ontology extraction is close to ours, they do use the created so far ontology, while we aim at an initial creation of a topic map.

The idea of exploiting Wikipedia semantics has been in the focus of several studies in the last years. The authors of [10] discuss semantic relationships in Wikipedia and the use of link types for search and reasoning. Recent research has also shown that Wikipedia can be successfully employed for NLP tasks, e.g. question answering [11] or text classification [12]. However, we are not aware of any research for using Wikipedia semantics in educational systems.

In the educational computing field, Kay and colleagues have been working on extracting ontologies from dictionaries, see for example [13], Henze and colleagues

focus in [14] on metadata generation from documents using context, while the authors of [15] utilize resource description formats for generating hypermedia structures.

The main difference between the above approaches and our approach is in the intended users of the extracted information: in our case it is intended for *human consumption*. The novelty of our approach is the collaborative building of the ontology by the author and the "helper agent". The agent extracts on demand topical structures that the author evaluates and possibly includes in the ontology. In this scenario the user interface and the presentation of the extracted topical structure are of equal importance compared to the accuracy of concept extraction. All this sets a new research perspective in the area of intelligent educational systems.

Various approaches, as described here, are applicable to the TM authoring support. At one pole are shallow techniques for extracting rather weak structural information while at the other extreme are deep statistical methods or knowledge driven parsers able to build rich semantic structures. We have shown that the proposed heuristics enable extracting structured information from HTML documents with usable degree of accuracy. We also succeeded to make use of Wikipedia without deep language understanding. This was made possible by applying standard techniques to match the structures with relevant properties. Empirical evaluation confirmed the value of encyclopedic knowledge for indexing of topics. The heuristics based on syntax alone, however, are not enough to capture the mass of structured information on the web. We are exploring the use of domain knowledge, through ontologies, for better topic extraction algorithms.

References

[1] Dicheva D., Dichev C.: Authoring Educational Topic Maps: Can We Make It Easier? 5th IEEE Intl Conf on Advanced Learning Technologies, July 5-8, 2005, Kaohsiung, Taiwan, (2005) 216-219
[2] Dicheva, D. & Dichev, C.: TM4L: Creating and Browsing Educational Topic Maps, British Journal of Educational Technology - BJET (2006) 37(3): 391-404
[3] ISO/IEC 13250:2000 Topic Maps, www.y12.doe.gov/sgml/sc34/document/0129.pdf
[4] Hodgson, J.: Do HTML Tags Flag Semantic Content?. IEEE Internet Computing, 5 (2001) 25
[5] Hepp M., D. Bachlechner, and K. Siorpaes. Harvesting Wiki Consensus - Using Wikipedia Entries as Ontology Elements. 1st Workshop Semantic Wiki 2006 at ESWC 2006, Budva, Montenegro (2006)
[6] Vargas-Vera, M., Motta, E., Domingue, J., Shum, S.,. Lanzoni, M.: Knowledge Extraction by using an Ontology-based Annotation Tool: kmi.open.ac.uk/projects/akt/publication-pdf/vargas-saw.pdf, 2000.
[7] Liu, B., Chin, C.W. and Ng, H. T.: Mining Topic-Specific Concepts and Definitions on the Web. Proc of WWW 2003, www2003.org/cdrom/papers/refereed/p646/p646-liu-XHTML/p646-liu.html (2003)
[8] Buitelaar, P., Olejnik, D. and Sintek, M.: A Protégé Plug-In for Ontology Extraction from Text Based on Linguistic Analysis. 1st European Semantic Web Symp.. http://dfki.de/~paulb/esws04.pdf (2004)
[9] Fortuna, B., Mladinic, D., Grobelnik, M: System for Semi-Automatic Ontology Construction. Proc. of 3rd European Semantic Web Symp.. http://www.eswc2006.org/demo-papers/FD18-Fortuna.pdf (2006)
[10] Krotzsch M, Vrandecic D., and Volkel M. Wikipedia and the Semantic Web -The Missing Links. In Proc. of Wikimania 2005, First International Wikimedia Conference, Wikimedia Foundation, 2005.
[11] Ahn D, Jijkoun V, Mishne G, Müller K, Rijke M, Schlobach S. Using Wikipedia at the TREC QA Track. In Proceedings of TREC 2004.
[12] Gabrilovich E., Markovitch S. Overcoming the Brittleness Bottleneck using Wikipedia: Enhancing Text Categorization with Encyclopedic Knowledge. Proc of AAAI 06, pp. 1301-1306, Boston, MA 2006.
[13] Apted, T., Kay, J., Lum, A. Supporting metadata creation with an ontology built from an extensible dictionary. 3rd Int. Conf. on Adaptive Hypermedia and Web-based Systems, Springer, (2004) 4-13.
[14] Henze, N., Dolog, P., & Nejdl, W., "Reasoning and Ontologies for Personalized E-Learning in the Semantic Web," Educational Technology & Society, 7 (4), (2004) 82-97
[15] Cardinales, K., Meire, M., Duval, E.: Automated Metadata Generation: the Simple Indexing Interface, Int. WWW Conference, Chiba, Japan, (2005)

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

VL-PATSy: Facilitating vicarious learning via intelligent resource provision

Richard Cox *, Jianxiong Pang
Department of Informatics, University of Sussex, Brighton, UK

Abstract. This paper describes an adaptive system called VL-PATSy, an extension to an existing system (PATSy) that adds a mechanism for serving vicarious learning (VL) resources. Vicarious learning is the notion that people can and will learn through being given access to the learning experiences of others. The VL resources represent an innovative form of learner support for a complex cognitive task (clinical reasoning). The VL resources are delivered in response to system-detected reasoning impasse events or in response to student demand in timely and context-sensitive ways. A three tier XML-oriented, rule-based architecture was chosen as an efficient and lightweight implementation solution and a design pattern approach was utilised.

Keywords.
vicarious learning, clinical reasoning, clinical education, event notification, design patterns, case-based teaching, intelligent tutoring, adaptive systems, metadata

1. Introduction

This paper describes a rule-governed, adaptive, educational support system and its associated core pedagogical methodologies in the clinical domain of speech and language therapy (SLT). The system has been developed as part of a project called VL-PATSy[1]. The overall aim of VL-PATSy is to provide vicarious learning resources to trainee Speech and Language Therapists (SLT's). Vicarious learning is the notion that people can and will learn through being given access to the learning experiences of others [1,4,13]. Simple traditional examples are instances such as master classes in music, and the process of clinical teachers going through cases with students. In VL, one student is the focus of tutorial attention, but others present also benefit as a result of observing the interaction. Other learners facing similar problems will pose issues within a similar "zone of proximal development", fostering an empathy which allows the vicarious learner to key into the acquisition process. VL offers an opportunity to preserve conversational elements and dialogue in e-learning contexts i.e. important components of Laurillard's 'conversational framework'[10]. VL also has potential for facilitating professional enculturation. The idea is that dialogue with expert practitioners helps induct the learner into the forms of discourse characteristic of the profession. In our project, the VL resources take the

*Correspondence to: Richard Cox, Department of Informatics, University of Sussex, Falmer, Brighton, BN1 9QH, UK. Tel.: +44 1273 67 8605; Fax: +44 1273 87 7873; E-mail: richc@sussex.ac.uk
[1] www.tlrp.org/proj/phase111/cox.htm

form of a database of video clips stored, indexed and disseminated as a new resource added to an established e-learning system (PATSy[2]). The database, together with other architecture modules, form VL-PATSy. The VL resource repository consists of a large (approx. 180) set of digital video clips. Each clip is a recording of either 2 students or a student and a tutor engaged in dialogue. Each video clip records a discrete clinical reasoning event (e.g. how to decide what diagnostic test to administer next). The participants in the VL video clips are interacting with virtual patient cases on the PATSy system [3,11,12]. The idea is that if a student using PATSy demonstrates inappropriate diagnostic reasoning or becomes stuck (reaches a reasoning impasse[3]), the system detects it and offers a choice of relevant VL resources (Figure 1).

2. The PATSy e-learning system

PATSy is a web-based generic shell designed to accept data from any discipline that has cases. The disciplines represented on PATSy currently include developmental reading disorders, neuropsychology, neurology/medical rehabilitation and speech and language pathologies. Sixty-one data-rich and extensively described cases of adults and children with disorders are accessible under 4 domain headings - speech and language, dyslexia, medical rehabilitation and neuropsychology. The system is used by more than 30 university departments including 80% of UK Speech and Language science departments. PATSy is designed for use in conjunction with more traditional methods of clinical training, professional education and academic teaching about disorders. As a multimedia database (video, audio, images), it serves as an electronic archive for case-based research as it contains rich collections of test data. It offers health science students an opportunity to practice diagnostic reasoning on 'virtual patients'. Students can access real patient data in the form of videos, assessments, and medical histories. PATSy's virtual patients can have psychometric and cognitive assessment tests 'administered' to them by learners. Typically, PATSy is used in blended learning contexts. Students attend clinical placements and tutors integrate online PATSy sessions with lecture content. PATSy complements taught content and clinical placements in important ways. Tutors are provided with a way of exposing a class of students to the same cases (clinical placements don't always expose students to a wide range of cases and student experiences on placements can vary widely). PATSy also contains published and rare cases that can bring the journal literature alive for students. Lecturers can 'hold back' PATSy cases. Students can then be assessed dynamically and authentically as they diagnose a previously-unseen case using PATSy in its exam mode.

PATSy has been designed to meet the needs of both students in initial clinical education (ICE) as well as by professional clinicians undergoing continuing professional development (CPD). This has been achieved by a learner-centred approach to the design of PATSy's modes of use. Clinical reasoning is difficult to teach via direct instruction and it is difficult to give students a set of rules that they may universally apply [9,16]. For this reason clinical science education (like law and other professions) is tending to adopt case-based teaching methods. Knowledge of the subject is represented in the form

[2]www.patsy.ac.uk

[3]A clinical reasoning impasse is defined in [7] as 'an inability to apply formal knowledge in a clinical setting'.

of many example cases [8,15]. PATSy is particularly well-suited to case-based teaching and for use in problem-based learning contexts. Case-based reasoning requires extensive experience and a large case-base for effective use. PATSy helps students acquire a 'critical mass' of cases and scaffolds them in their transition from predominantly hypothetico-deductive modes of reasoning to more heterogeneous, expert-like modes. In PATSy's ICE modes, students are encouraged to reason in hypothetico-deductive mode to a greater extent than is the case for experienced clinicians (e.g. CPD users). Tutors may configure PATSy such that ICE users are discouraged from top-down test 'score pattern spotting' whereas CPD users are free to do so.

PATSy has built-in student-system interaction recording, making it an educational research instrument for the microgenetic study of students' clinical reasoning skill acquisition. The log records in detail which tests are accessed and the order in which they were 'administered' to virtual patients.The system has been used to study students' and experts' diagnostic reasoning. A range of reasoning difficulties has been identified and a 6-level diagnostic accuracy scheme has been developed [2,6,7]. PATSy can also be configured to periodically prompt students with questions such as: 'What conclusions can you draw from what you have just seen?', 'Where do you want to go next (further test data?, introductory video?)', 'Why?' Students' typed responses to these prompt questions are interleaved with the system-generated log data. As mentioned above, the PATSy system has recently been extended by the addition of VL support resources.

Figure 1. VL-PATSy suggests a range of video clips to a student using PATSy following its detection of a clinical reasoning impasse or event. The PATSy user has chosen a video clip of two students discussing the clinical reasoning topic 'heuristics for generating hypotheses'.

Development of VL resources to support students' clinical reasoning

Task-directed discussion (TDD) exercises were used as a methodology for generating pedagogically effective dialogues for use as vicarious learning resources. TDDs are language games of a type commonly used in English-as-a-second-language (ESL) teaching

[14]. A TDD is a discrete language activity with a set goal [13]. They are useful as a means of generating discussion where spontaneous discussion is infrequent and where capturing naturally occurring dialogue is difficult and uncontrollable. Compared to open classroom or small-group tutorial contexts, the participation by pairs of participants in TDD exercises is designed to ensure a high degree of topic focus and increased amounts of discussion, together with deeper explorations of domain concepts. A typical domain-specific TDD topic might be one which elicits discussion about the definition of a concept. (e.g. *'Please provide a definition of paraphasia'*) or comparisons between concepts (*'Discuss when the Howard and Franklin picture reading test might be a better choice than the ADA reading-of-words test in diagnosing comprehension difficulties in an aphasic patient'*). Other TDD examples focus on the use of professional register, e.g. *'Read the following description of a client's communication difficulties and then rephrase it in professional terminology'*.

An extensive collection of TDD-derived dialogues (approx. 180) have been recorded, edited and incorporated into a structured database. VL-PATSy is a mixed initiative system. Students may browse VL-PATSy's VL resources at any point as they work through a case. The video clips are tagged with metadata and the learner's interaction with PATSy can fire rules which result in students being offered dialogues to view when they strike a reasoning impasse. The system monitors students' interactions and intervenes where necessary. Event-condition-action (ECA) rules fire if certain events are detected. To illustrate, if a student administers an unusual and inappropriate sequence of language tests to a virtual patient, then a rule fires and VL video clips are proactively offered on relevant topics such as 'hypothesis testing'. Clips with content tags such as '*a student and tutor discuss how understanding what a test does helps with planning a test strategy*' and '*a student and tutor discuss confirming and disconfirming a hypothesis*' might be offered (e.g. Figure 1). In research currently underway, VL-PATSy is being used as a research test-bed for comparing the relative efficacy of VL clips consisting of either student-tutor or student-student dialogue.

3. The architecture of VL-PATSy

The requirements of the VL-PATSy system are for an architecture exhibiting a high degree of independence and separation from the existing PATSy system. Hence a primary consideration of the design for VL-PATSy was for it to run separately from the existing PATSy system and for it to maintain its links to PATSy via message exchange. It was also desirable that the system should be resilient, scalable, extensible and maintainable. A three-tier architecture was employed. This consists of (1) a frontend tier i.e. an information, or content presentation layer, (2) a middle tier i.e. a monitor and notification mechanism, and (3) a backend tier in the form of multi-role server components and persistence facility (Figure 2). Three-tier architectures are common in many web-based applications and allowed us to choose from a wide range of existing tools and design patterns. They are well-suited to a rule-based architecture - a rule-engine is by nature a server-class software component. Three-tier architectures have been in use in several different domains. For example, in the enterprise computing area, the 3-tiers consist of the client side (web browser or proprietary client), middleware services (messaging, persistence, security, caching and event notification) and a backend (i.e. server-side components).

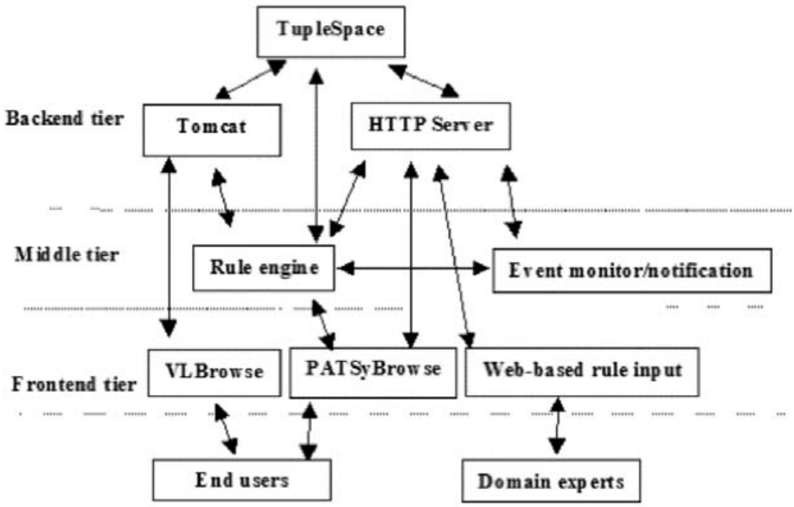


Figure 2. The 3-tier architecture of VL-PATSy system.

However, VL-PATSy's architecture differs from these examples in several key respects. The Server can extend its services by allowing domain experts (expert speech and language clinician educators in our case) to add more rules to the rule store. Domain experts can define functionality via rules. The front end and rule server components apply context information. The rule engine works with the HTTP service but is independent of the underlying web application. The architecture makes the underlying HTTP transparent to the domain experts. It enables domain experts to enter functionality and domain-specific knowledge into the tutoring system and provides communication channels between rule servers that can be used for interaction handling. The front end tier represents the chosen web application and HTTP communication facility. It is a combination of the presentation layers of both the PATSy and VL-PATSy systems. Three features of the front end tier are: (1) The rule servers are provided with frequent messages about the learner's server requests as soon as they arrive at the PATSy server, (2) VL-PATSy's responses (e.g. to present information to the learner) are not driven by user request directly, but indirectly - the information that is presented is driven by the learner's activities, but not by the learner's direct requests. Finally, (3) low-level learner behaviours (as recorded by PATSy's user-system logs) are mapped onto more abstract ECA events. For example, a student may indicate premature commitment to a very specific hypothesis and begin his or her testing of the patient with a narrowly focussed test instead of starting out with a broad test. An abstract ECA rule might correspond to the statement 'begin diagnosis using broad tests, and subsequently test specific hypotheses with more narrowly-focussed tests'. For each of the 28 virtual patient cases on PATSy, there would be many particular test sequences that might fire this abstract rule.

Note that a major requirement was for the 'push' of pedagogical advice from the VL-PATSy system to run concurrently and asynchronously, without blocking or interrupting the PATSy system's normal mode of operation. The middle-tier is a monitoring and notification mechanism (i.e. a 'spy'). One of the most important interactions between PATSy and VL-PATSy occurs via the processing of log data. The middle-tier must mediate between PATSy and VL-PATSy in order to relate the user-system interaction log data

from PATSy to the delivery of vicarious learning content by VL-PATSy. Two crucial constraints are imposed on the middle-tier - (1) The rule servers must be informed of events that happen in each learning session and (2) PATSy must run normally without noticeable loss of performance. VL-PATSy responds to particular student-PATSy interaction events. If an event matches certain conditions then actions result. The actions cause an appropriate subset of VL resources (annotated video clips) to be offered to the student.

ECA rules have been used for defining the mapping relation between learner behaviours and teaching content delivery using XML standards. The interactions with VL-PATSy must be transparent to the PATSy users. This means that VL-PATSy must not interfere with event routing or execution in the PATSy system. PATSy's user-system event logging system dynamically builds a publisher/subscriber design pattern in the rule engine. Rule objects are subscribers and they can subscribe and de-subscribe to PATSy events in run time. These design decision have proved successful in a number of experimental runs of the system. VL-PATSy's backend tier components include a database module, which is responsible for the management of metadata for video clips and ECA rules written in XML. The database module is based on a database-featured communication middleware called a 'tuple space'. PATSy's test knowledge base has been enriched by the addition of rigorous descriptions of tests and test items in the form of Test Description Language (TDL, [1]). The VL video clips are represented richly in terms of case, type of dialogue participant, clinical reasoning issue, etc via an XML-compliant Video Description Language (VDL).

3.1. VL-PATSy rules

The rule specification language is designed to have adequate expressibility for SLT clinical reasoning using XML mark-up language. An example of a pedagogical rule for detecting and responding to a student's test knowledge impasse might be 'when a student moves from test X to test Y, but X→Y is an inappropriate testing sequence, then offer relevant VL video clips' (Figure 3).

```
<?xml version=1.0 encoding=UTF-8?>
<rule owner=vlpatsy applies_to=all id=TESTS_BAD_FOCUS
     enabled=true changed=2006-08-26T13:49:00>
<when>
  <rep match='Test' times='2' checkIf='isNotFocus'></rep>
</when>
<do name='DS-HT-TS' id='1' cause='TESTS_BAD_FOCUS'>
  <fetch klahr1='DS' klahr2='HT' usage='TS'></fetch>
</do>
</rule>
```

Figure 3. : The 'TESTS_BAD_FOCUS' rule in VL-PATSy

Each rule is composed of a trigger block and an action block. A basic format of a rule is:

when <some situation occurs> do <something>

Of course, VL-PATSy's pedagogical rules may conflict and conflict resolution may be required. All rules are divided into two groups, group one is called 'top priority' rules

- rules in this group must be fired if trigger conditions hold; group two is called 'normal priority' rules, if several rules fire simultaneously, then one is chosen randomly and the others are discarded[4] Event patterns as triggers are an important data type that represent learner's behaviours. In the case of Figure 3 the TESTS_BAD_FOCUS rule is assigned 'normal priority'. When a learner selects two tests X and Y, the TESTS_BAD_FOCUS rule is fired if and only if Y is not a (clinically) reasonable subsequent test to X. The 'action' block in Figure 3 contains an instruction to deliver a list of vicarious learning video clips. For example, TESTS_BAD_FOCUS rule specifies that those video clips should be tagged as DS (domain specific), and HT (hypothesis testing) and TS (deliver after a series of tests has been administered to the patient).

4. Related work

In intelligent-tutoring (ITS) terms, VL-PATSy is similar to a model-tracing tutor - e.g. Ms. Lindquist [5], an algebra tutor. It has a student model and a tutorial reasoning model that produces a human-like interactive dialogue to help students learn how to write algebraic expressions for word problems. The remediation feature is similar to the failure advisor rules of VL-PATSy. Ms. Lindquist utilises a continuous running 'conversation' and its data structures have been designed around a need for dialogue. In VL-PATSy, this kind of dialogue is less important, and VL-PATSy focusses more on monitoring the student's activities and responding with suitable VL offerings in the form of links to relevant VL video clips. Interestingly, Ms Lindquist has a separate tutorial model that encodes pedagogical content knowledge in the form of different tutorial strategies and this module was partially developed by observing an experienced human tutor. This feature is in many senses similar to VL-PATSy's design idea of pedagogical rules in that each cause-result relation has been specified by domain experts on the basis of their clinical expertise. However, unlike VL-PATSy, and given the non-trivial task of setting up a constraint/rule base, Ms.Lindquist relies heavily on authoring tools. To use Ms Lindquist users need to learn how to use its authoring tool whereas VL-PATSy provides a lighter-weight rule authoring system for its domain experts to use.

5. Conclusions

An adaptive system called VL-PATSy has been described. It represents an extension to an existing system (PATSy) which adds a mechanism for intelligently serving vicarious learning (VL) resources. The VL resources are an innovative form of learner support during complex cognitive tasks (clinical reasoning). They are delivered a) in response to system-detected learning events or b) on student demand in timely and context-sensitive ways. A three tier XML-oriented, rule-based architecture was chosen as an efficient and lightweight solution.

[4] A major source of rule-firing conflict tends to happen when students type reflective notes into PATSy's log. For example, two rules might fire due to a) the phrasing of the entry failing to demonstrate the use of professional register and b) the entry consisting of fewer than 200 characters.

Acknowledgements

The support of the Economic and Social Research Council is acknowledged (ESRC TLRP grant RES-139-25-012). Thanks to colleagues Professor Rosemary Varley, Dr John Lee, Dr Julie Morris, Kirsten Hoben, Dr Barbara Howarth & Susen Rabold.

References

[1] Cox, R. (2005) Vicarious learning and case-based teaching of clinical reasoning skills. ESRC TLRP Seminar Series on *Contexts, communities, networks: Mobilising learners' resources and relationships in different domains.* Open University, Milton Keynes, October. Paper available at `www.tlrp.org/dspace/`

[2] Cox, R., Hoben, K., Howarth, B., Lee, J., Morris, J., Pang, J., Rabold, S. & Varley, R. (2005) Clinical reasoning skill acquisition: identifying learning issues and developing vicarious learning resources, *T*eaching and Learning Research Programme Annual Conference, (University of Warwick, November). Available via 'Publications' link at: `www.tlrp.org/proj/phase111/cox.htm`

[3] Cox, R., & Lum, C. (2004) Case-based teaching and clinical reasoning: Seeing how students think with PATSy. In Brumfitt, S. (Ed.) *Innovations in professional education for Speech and Language Therapists*, London: Whurr.

[4] Cox, R., McKendree, J., Tobin, R., Lee, J. & Mayes, T. [1999] Vicarious learning from dialogue and discourse: A controlled comparison. *Instructional Science*, **27(6)**, 431–457.

[5] Heffernan, N. T., & Koedinger, K. R. (2002) An Intelligent Tutoring System Incorporating a Model of an Experienced Human Tutor. *Proceedings of the Sixth International Conference on Intelligent Tutoring Systems*, Bierritz, France.

[6] Hoben, K., Varley, R., & Cox, R. (to appear) The clinical reasoning skills of speech and language therapy students. *International Journal of Communication and Language Disorders.*

[7] Howarth, B., Hoben, K., Morris, J., Varley, R., Lee, J. & Cox, R. (2005) Investigating students' clinical reasoning: Analysing think-aloud protocols and integrating multiplesource data. *Proceedings of the 11th Biennial Conference of the European Association for Research in Learning and Instruction (EARLI)*, Cyprus, August.

[8] Irby, D.M. (1994) Three exemplary models of case-based teaching. *Academic Medicine*, **69**, 947-953.

[9] Jonassen, D.H. (1996) Scaffolding diagnostic reasoning in case-based learning environments. *Journal of Computing in Higher Education*, **8(1)**, 48-68.

[10] Laurillard, D. (1993) *Rethinking University teaching: A framework for the effective use of educational technology.* London: Routledge.

[11] Lum, C., Cox, R., Kilgour, J., Morris, J. & Tobin, R. (1999) PATSy: A multimedia, web-based resource for aphasiologists and neuropsychologists in research and education. *Aphasiology*, **13**, 573-579.

[12] Lum, C. & Cox, R. (1998) PATSy - A distributed multimedia assessment training system. *International Journal of Communication Disorders*, **33**, 170–175.

[13] Mayes, J.T., Dineen, F., McKendree, J. & Lee, J. (2001) Learning from watching others. In C. Steeples & C. Jones (Eds.), *Networked learning: Perspectives and issues.* London: Springer.

[14] Skehan, P. (1988) *A cognitive approach to language learning*. Oxford: Oxford University Press.

[15] Whitworth, A., Franklin, S. & Dodd, B. (2004) Case based problem solving for speech and language therapy students. In Brumfitt, S. (Ed.) *Innovations in professional education for Speech and Language Therapists*, London: Whurr.

[16] Williams, S.M. (1993) Putting case-based learning into context: Examples from legal, business and medical education. *Journal of the Learning Sciences*, **2**, 367-427.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

‘Tis Better to Construct than to Receive? The Effects of Diagram Tools on Causal Reasoning

Matthew W. Easterday, Vincent Aleven, Richard Scheines
Human-Computer Interaction Department, Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh, PA, 15213-3891

Abstract Previous research on the use of diagrams for argumentation instruction has highlighted, but not conclusively demonstrated, their potential benefits. We examine the relative benefits of using diagrams and diagramming tools to teach causal reasoning about public policy. Sixty-three Carnegie Mellon University students were asked to analyze short policy texts using either: 1) text only, 2) text and a pre-made, correct diagram representing the causal claims in the text, or 3) text and a diagramming tool with which to construct their own causal diagram. After a pretest and training, we tested student *performance* on a new policy text and found that students given a correct diagram (condition 2 above) significantly outperformed the other groups. Finally, we compared *learning* by testing students on a third policy problem in which we removed all diagram or tool aids and found that students who constructed their own diagrams (condition 3 above) *learned* the most. We describe these results and interpret them in a way that foreshadows work we now plan for a cognitive-tutor on causal diagram construction.

Keywords: External representations; Diagrammatic reasoning; Causal reasoning

Introduction

To become effective citizens, students must be able to analyze public policy. For example, students should be able to reason about *causal questions* such as: “Will decreasing the amount of junk-food commercials on TV decrease childhood obesity?” Furthermore, students’ reasoning must account for different claims presented by different sources, e.g. “Researchers claim that watching junk food commercials causes obesity, while industry advocates argue that junk food commercials only affect the brand of junk food consumed, not the total number of calories.”

Taken together, several bodies of research suggest that while students have difficulty reasoning about causal arguments [1-3], diagrammatic representations [4-12] and tools [13-16] might improve their reasoning. However, as Ainsworth [5] notes,

> ... research on the benefits of providing learners with more than one representation has produced mixed results. For example, a number of studies have found that learners benefit from either constructing or being presented with [diagrams]... Unfortunately, just as many studies have shown that learners can fail to benefit from these proposed advantages of [diagrams]...

Furthermore, the efficacy of a given diagram format interacts heavily with both the particular task on which the diagram is applied, and students’ familiarity with the diagram format, i.e. the fact that a concept map can improve students’ recall [17] does not necessarily imply that a causal diagram will improve the accuracy of students’ policy inferences. Thus, the following are important open questions:

- For what domains and with what kind of pedagogy do we think diagrams will help? For example, should we use diagrams to teach causal reasoning about public policy texts? Given student fluency with text in general, we may not be

able to design pedagogy with diagrams that significantly improve reasoning in comparison with text, without exorbitant cost. Ainsworth [5] shows the difficulty of learning a new diagram format in general, and only a few studies (such as Suthers & Hundhausen [15]) have examined diagrams for causal reasoning, usually focusing on science, not policy.

- Should we give students pre-made diagrams, or should they construct their own? Some argue that students come to a *deeper understanding* of a problem or task constructing their own representations of it, while others argue that diagrams constructed by students contain too many errors to be useful, or that the empirical evidence for the benefit of construction is scant [14]. Many studies either do not examine (or do not find benefits for) "pure" diagram construction relative to "pure" interpretation, because they supplement construction with feedback [18] or scaffolding [19, 20], do not provide a correct diagram for the interpretation group [21], or do not directly contrast construction and interpretation [18].
- Does using/constructing a diagram have the same effect on learning as it does on performance? Even if we can temporarily increase performance by using diagrams, students may not learn much in transfer tasks in which the correct diagram is not immediately available.

1. Task and Intervention

To explore these questions, we gave students short, fictional, policy texts (Figure 1).

Childhood obesity is now a major national health epidemic. A number of facts are widely agreed upon by the public and scientific community: exercise decreases obesity, and eating junk food increases obesity. It's also clear that people who watch more TV are exposed to more junk food commercials.

Parents for Healthy Schools (PHS), an advocacy group which fought successfully to remove vending machines from Northern Californian schools, claims that junk-food commercials on children's television programming have a definite effect on the amount of junk food children eat. In a recent press conference, Susan Watters, the president of PHS stated that "...if the food companies aren't willing to act responsibly, then the parents need to fight to get junk food advertising off the air."

A prominent Washington lobbyist Samuel Berman, who runs the Center for Consumer Choice (CCC), a nonprofit advocacy group financed by the food and restaurant industries, argues that junk food commercials only "influence the brand of food consumers choose and do not not affect the amount of food consumed." While Mr. Berman acknowledges that watching more TV may cause people to see more junk food commercials, he remains strongly opposed to any governmental regulation of food product advertising.

Recent studies by scientists at the National Health Institute have shown that watching more TV does cause people to exercise less.

Figure 1. Policy text on obesity.

Our tasks involved answering questions like: "According to the PHS, will making kids exercise more reduce the number of junk food commercials they watch?" Note that neither *junk food commercials* nor *exercise* affects the other, so the correct answer is "no."

What is the correct diagram for a policy text and why? In the last two decades, researchers have produced a rigorous theory of causal reasoning that rests primarily on a semantics for causal diagrams in the form of directed graphs [22, 23]. Because the ability to use these diagrams has implications for reasoning in general and for policy

reasoning in particular, teaching the theory has become somewhat of a priority. But even after weeks of instruction, students in causal reasoning courses, like those in other formal domains like algebra or physics, still fall short of being able to build and utilize formal representations reliably and accurately. This led us to wonder whether the cognitive cost of teaching causal diagrams outweighs the presumptive benefit of using or building causal diagrams from texts like Figure 1.

To test differences in performance and learning between students who used diagrams, diagramming tools, or neither (only text representations), we randomly assigned students who had no prior training in causal reasoning to one of three conditions:

1. **Text** students (the control group) received all case studies as text only (Figure 1).
2. **Diagram** students received, in addition to a text version, a correct, diagrammatic representation of the case study (Figure 2).

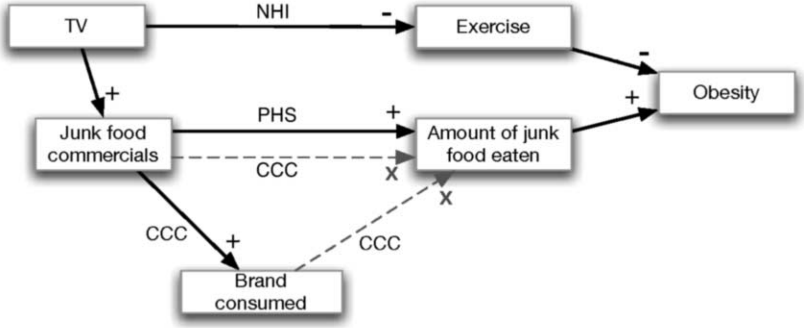


Figure 2. A causal diagram representing the case-study on obesity. Boxes represent causal variables, and arrows represent either positive (+), negative (-), or no (x) influence of one variable on another. An annotation on the arrow (e.g. PHS) identifies the source making the causal claim.

Note that to solve the question about exercise reducing the amount of junk food eaten using the diagram in Figure 2, students only need to notice that there is no set of arrows leading from the exercise variable to the amount of junk food eaten variable.

3. **Tool** students received the case study along with a computer tool with which they could construct their own diagrams (Figure 3).

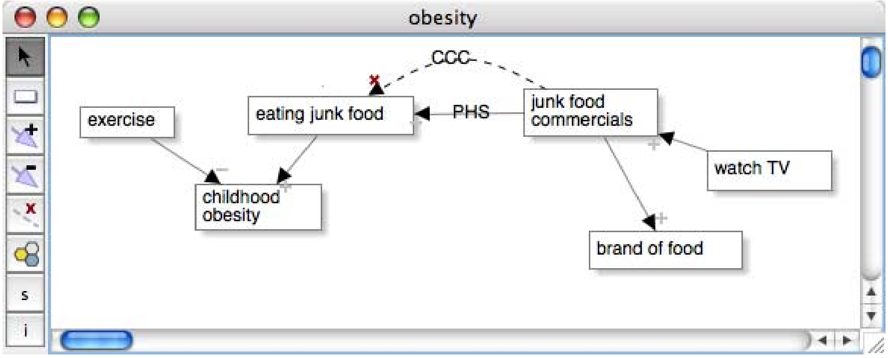


Figure 3. The iLogos tool shown in causal mode [24].

2. Participants and Setting

We investigated these questions with 63 Carnegie Mellon University students enrolled in undergraduate philosophy classes but who had no prior training in causal reasoning.

3. Research design

The study consisted of a brief, 15 minute training (M = 16.32, SD = 5.43), and three tests in which students were given a policy text and asked to answer 10 causal reasoning questions, (see Figure 4.) All students began the experiment with a policy argument on the environment (**pretest**) presented in text only. After the pretest, text students received training on causal reasoning without diagrams, while diagram and tool students received training with diagrams. We then tested *performance* (**performance test**) by giving students another policy argument on obesity presented as text to the text students, presented as text with a diagram to the diagram students and presented as text with the diagramming tool to tool students. Finally, to test *learning*, all students were given a third text on crime in which policy arguments were presented as text only (**learning test**).[1]

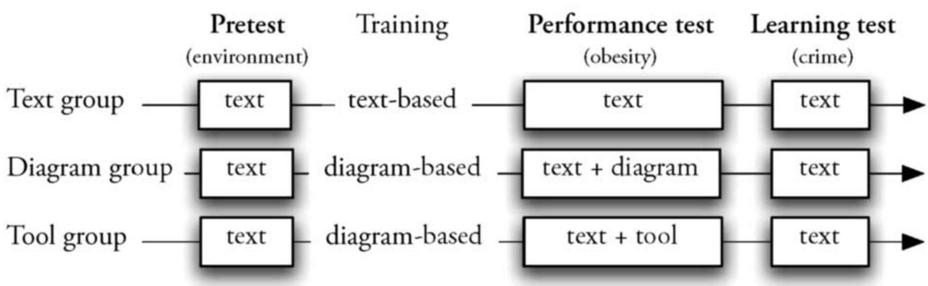


Figure 4. The experimental procedure showing order of tests and training for each group.

In the training, both groups received 4 interactive web-pages of instruction on causal reasoning. The diagram and tool groups' instruction included diagrams (as in Figure 2), while the text group's did not. To make the training as close to identical as possible, every diagrammatic explanation in the diagram/tool training was matched by an equivalent prose explanation in the text training. Tool students also received an additional page of instruction describing how the buttons of the tool worked (Figure 3), but with no additional information about diagrams or reasoning. While students received feedback on the problems they solved on the training, we gave them no feedback about their answers on the tests.

4. Data collection

On each test, students were asked 10 multiple choice, causal questions (e.g. "According to the PHS, will making kids exercise more reduce the number of junk food commer-

[1] Note that the order of policy texts in the tests was not counter-balanced, i.e. all students received the policy text on environment in the pretest, followed by obesity on the performance test, followed by crime on the learning test. The underlying causal structure of each policy text however, (i.e. the number of variables and connections between variables), was identical across policy texts–texts differed only in cover story.

cials they watch?"). Students could answer one of three ways, either that: a) there would be a causal effect (e.g. reducing junk food commercials would affect obesity), b) there would be no causal effect, or c) there was not enough information to decide. We also recorded the time students spent on each test, and whether or not they constructed a diagram on scratch paper.

5. Results

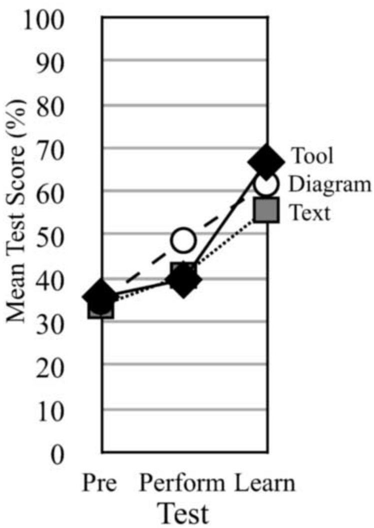


Figure 5. Mean test scores for text (n = 24), diagram (n = 24) and tool (n = 15) students on pre-, performance and learning tests.

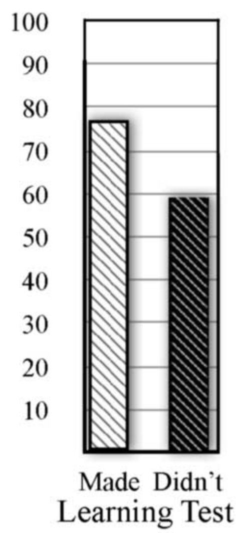


Figure 6. Mean test scores on the learning test for students who made (n = 6), or didn't make (n = 57) diagrams.

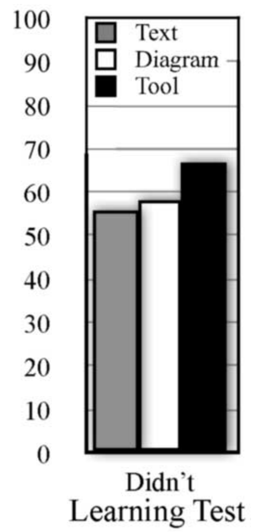


Figure 7. Mean test scores on the learning test for the students who didn't make diagrams (n = 57).

Overall performance and learning results. As can be seen in Figure 5, all groups performed at chance on the **pretest** (Text = 34%, Diagram = 35%, and Tool = 36% with no significant differences between groups). After training, students were given a performance test in which policy information was presented as text to text students, as text with a correct diagram to diagram students, and as text with a diagramming tool to tool students. On this **performance test**, diagram students scored 49%, outperforming both the text students who scored 41% ($p < .05$, $df = 54$) and the tool students who scored 40% (although the difference was not significant) showing that students reason better when text is accompanied by a correct diagram, rather than by a tool or alone.[2] On the **learning test** however, in which all students received policy information as text only, *both* tool students (67%) and diagram students (62%) outperformed text students (56%, $p < .05$, $p = .07$ respectively, $df = 57$), so while having a correct diagram seems to provide a superior advantage for improving *performance*, the dramatic gains of the tool group on the learning test undercut any clear advantage of having a diagram for *learning*.

[2] We performed a multiple regression with the mean test score as the outcome variable, 2 dummy variables representing experimental condition, and time on test and training as covariates. On the learning test, we used time on training as a covariate. The multiple regression analysis is equivalent to an ANOVA and was used because the coefficients are easier to interpret.

Effect of making a diagram on the learning test. To better understand the tool group's learning gains, we looked separately at students who made diagrams and students who didn't make diagrams on the learning test (see Figure 6). The 6 students who made diagrams performed better (77%) than the 57 students who did not make diagrams (59%, $p < .05$, $df = 61$) demonstrating either the usefulness of diagrams or a selection effect showing that "good" students make diagrams.

Effect of tool practice on the learning test. Figure 7 shows the scores of students who did *not* make diagrams on the learning test, (13 tool students, 20 diagram students, and 24 text students). Among these students, students in the tool condition who had practiced making diagrams on the performance test scored 67%, outperforming the students in the text condition who scored 56% ($p < .05$, $df = 53$), and students in the diagram condition who scored 58% ($p < .10$, $df = 53$). Although these results are correlational they suggest that, *when diagrams are unavailable,* having practiced making diagrams with a tool leads to an increase in *learning.*

Time. There were no significant differences in time between groups on the pretest, performance test or learning test, however, the diagram group spent longer (16.1 min) on the shared training than the text group (13.4 min, $p < .05$, $df = 60$). One could argue that diagrams simply make students spend longer on training, however, that explanation cannot account for the results of the learning test or for the fact that the training did not have such an effect on the tool group, whose time on training (14.8 min) was not significantly different than that of the text group. In fact, given the dramatically greater amount of practice students have had *prior* to the experiment with text as compared to diagrams, it is surprising that differences in training time were not greater. Of far greater import is the relatively short time that *all* groups spent on training and testing–whereas professors and graduate students took approximately 1-1.5 hours to complete the experiment during pilot testing, participants took an average of only 32 minutes!

Main results. In summary, there are two results that require explanation: 1) diagram students showed better *performance* than text and tool students, and 2) tool students demonstrated greater learning.

6. Interpretation

How do we explain performance of the diagram students and the learning of the tool students? Using a diagram to make an inference requires at least three basic steps: 1. *comprehension*, 2. *construction,* and 3. *interpretation* (see Figure 8).

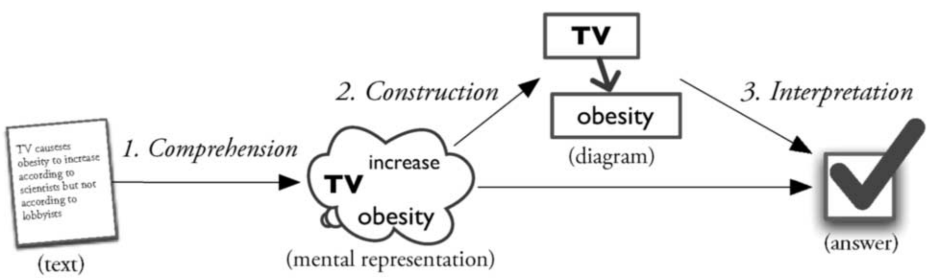


Figure 8. General steps in using an external representation.

In the *comprehension* step [25], students recognize a relevant piece of information in the source text, (e.g. a causal claim), presumably forming some corresponding mental representation. From that mental representation, students may either try solve the

inference problem directly, (which may impose a significant working memory burden), or they may externalize that information in a new form by *constructing* a diagram. Students then *interpret* the diagram, making the inference needed to solve the problem.

If the diagram format *works*, (i.e. it reduces the required amount of working memory or number of cognitive operations), and students have learned how to use the format, then using a diagram should be more effective than using text. Also diagrams should help more than tools, because when students use a tool they must *comprehend, construct* and *interpret,* whereas students using a diagram only have to *interpret.* The process in Figure 8 thus predicts the superior results of the diagram students on the *performance* test–students had greater success with diagrams than with text or tools.

The process in Figure 8 also explains the *learning* results of the tool students: if practicing constructing diagrams improves one's *comprehension* skills, then tool students should perform better than text and diagram students when diagrams aren't used, which is exactly what we see in Figure 7. Note, that even with improved comprehension skills, solving the problem from the mental representation should still be more difficult than solving the problem with a diagram (if the diagram format has been properly designed to reduce working memory burden or the number of cognitive operations), which are the results we see in Figure 6.

To summarize, when we tested reasoning *performance*, we saw that the diagram students outperformed the others, suggesting that, indeed, the *interpretation* step (of reading off the diagram to get the answer) is easier than the combined *comprehension*, *construction* and *interpretation* steps required when one has a tool, (a conclusion supported by the observation that no tool student was able to construct a correct diagram on the performance test). On the other hand, when we tested *learning* by removing all diagrams and tools, students who had practiced constructing diagrams with tools outperformed the others, suggesting that practice constructing diagrams may improve students' comprehension skills.

7. Conclusion

We found that after only 15 minutes of training, students were able to make almost 10-20% more accurate inferences about the effects of different social policies when they had a correct diagrammatic representation,[3] and that practice constructing diagrams also improved students' reasoning by approximately 10%.[4]

So it seems that diagrams, whether constructed or received, are indeed useful for (learning) causal reasoning about public policy. With effective instruction over a period of weeks, we are hopeful that students can not only handle the sorts of problems we examined here, but also policy problems of more realistic complexity. In pursuit of this goal, we are now conducting protocol studies of diagram construction on experts and novices. We hope to leverage these studies into a cognitive model of diagram construction, and leverage the cognitive model of diagram construction into a cognitive tutor.

---

[3] Comparing the average score of 49% for diagram students to the 40% and 41% of the text and tool students on the performance test, and comparing the average score of 77% for students that made any diagram on learning test to the 59% of students who didn't make diagrams.

[4] Comparing the averages of 67% of tool students who did not make diagrams on the learning test to the 58% of diagram students and 56% of text students.

References

[1] Scheines, R., Easterday, M., & Danks, D. (in press). Teaching the Normative Theory of Causal Reasoning. In A. Gopnik, & L. Schultz (Eds.), *Causal learning: Psychology, philosophy, and computation*. Oxford, England: Oxford University Press.

[2] Kuhn, D. (1991). *The skills of argument*. Cambridge, MA: Cambridge University Press.

[3] Voss, J. F., Perkins, D. N., & Segal, J. W. (1991). *Informal reasoning and education*. Hillsdale, NJ: Lawrence Erlbaum.

[4] Larkin, & Simon (1987). Why a diagram is (sometimes) worth ten thousand words. *Cognitive Science*, 11, 65-99.

[5] Ainsworth, S. E. (in press). DeFT: A conceptual framework for learning with multiple representations. *Learning and Instruction*.

[6] Harrell, M. (2004). *The improvement of critical thinking skills in What Philosophy Is* (Tech. Rep. No. CMU-PHIL-158). Pittsburgh, PA: Carnegie Mellon University, Department of Philosophy.

[7] Scaife, M. and Rogers, Y. (1996) External cognition: how do graphical representations work? *International Journal of Human-Computer Studies*, 45(2), 185-213.

[8] Cox, R. (1999) Representation construction, externalised cognition and individual differences. *Learning and Instruction*, 9(4), 343-363.

[9] Novick, L. R., & Hurley, S. M. (2001). To matrix, network, or hierarchy: That is the question. *Cognitive Psychology*, 42(2), 158-216.

[10] Mayer, R. E., & Moreno, R. (2002). Aids to computer-based multimedia learning. *Learning and Instruction*, 12, 107-19.

[11] Bauer, M. I., & Johnson-Laird, P. N. (1993). How diagrams can improve reasoning. *Psychological Science*, 4(6), 372-8.

[12] Pinkwart, N., Aleven, V., Ashley, K., & Lynch, C. (2006). Toward legal argument instruction with graph grammars and collaborative filtering techniques. *Proceedings of the 8th International Conference on Intelligent Tutoring Systems*, 227-36.

[13] Kirschner, P. A., Buckingham Shum, S. J., & Carr, C. S. (Eds.). (2003). *Visualizing argumentation: Software tools for collaborative and educational sense-making. London*: Springer.

[14] Van den Braak S. W., Van Oostendorp H., Prakken H., Vreeswijk G. (2006, August). *A critical review of argument visualization tools: do users become better reasoners?* Paper presented at the ECAI-06 Workshop on computational models of natural argument, Trento, Italy.

[15] Suthers, D., & Hundhausen, C. (2003). An Empirical Study of the Effects of Representational Guidance on Collaborative Learning. *Journal of the Learning Sciences*, 12(2), 183-219.

[16] Harrell, M. (2005). Using argument diagramming software in the classroom. *Teaching Philosophy*, 28(2), 163-77.

[17] Nesbit, C., & Adesope, O. (2006). Learning with concept and knowledge maps: A meta-analysis. *Review of Educational Research*, 76(3), 413-48.

[18] Cox, R. (1997). Representation interpretation versus representation construction: An ILE-based study using switchERII. *Proceedings of the 8th World Conference on Artificial Intelligence in Education*, 434-41.

[19] Hall, C., Bailey, J., & Tillman, C. (1997). Can Student-Generated Illustrations Be Worth Ten Thousand Words? *Journal of Educational Psychology*, 89(4), 677-81.

[20] Van Meter, P. (2001). Drawing construction as a strategy for learning from text. *Journal of Educational Psychology*, 93(1), 129-40.

[21] Grossen, B., & Carnine, D. (1990). Diagramming a logic strategy: Effects on difficult problem types and transfer. *Learning Disability Quarterly*, 13(3), 168-82.

[22] Spirtes, P., Glymour, C., and Scheines, R. (2000). *Causation, Prediction, and Search* (2nd ed.). Cambridge, MA: MIT Press.

[23] Pearl, J. (2000). *Causality: Models, Reasoning, and Inference*. Cambridge: Cambridge University Press.

[24] Easterday, M. W., Kanarek, J. S., & Harrell, M. (in press). Design Requirements of Argument Mapping Software for Teaching Deliberation. In T. Davies, & B. S. Noveck (Eds.), *Online Deliberation: Design, Research, and Practice*. Stanford, CA: CSLI Publications/University of Chicago Press.

[25] Koedinger, K. R., & Nathan, M. J. (2004). The real story behind story problems: Effects of representations on quantitative reasoning. *The Journal of the Learning Sciences*, 13(2), 129-64.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Evaluating Legal Argument Instruction with Graphical Representations Using LARGO

Niels PINKWART[1], Vincent ALEVEN[2], Kevin ASHLEY[3] and Collin LYNCH[4]
[1]Clausthal University of Technology, Computer Science Institute, Germany
[2]Carnegie Mellon University, HCI Institute, Pittsburgh PA, USA
[3]University of Pittsburgh, LRDC and School of Law, Pittsburgh PA, USA
[4]University of Pittsburgh, Intelligent Systems Program, Pittsburgh PA, USA

Abstract. Previous research has highlighted the advantages of graphical argument representations. A number of tutoring systems have been built that support students in rendering arguments graphically, as they learn argumentation skills. The relative tutoring benefits of graphical argument representations have not been reliably shown, however. In this paper we present an evaluation of the LARGO system which enables law students graphically to represent examples of legal interpretation with hypotheticals they observe while reading texts of U.S. Supreme Court oral arguments. We hypothesized that LARGO's graphical representations and advice would help students to identify important elements of the arguments (i.e., proposed hypotheses, hypothetical challenges, and responses) and to reflect on their significance to the argument's merits better than a purely text-based alternative. In an experiment, we found some empirical support for this hypothesis.

Introduction

Researchers in the field of intelligent tutoring systems have long been interested in developing systems that can help students learn through argument and debate or support the learning of argumentation skills [2, 7, 10].

One of the subtler aims of a law school education is to help students develop a "legal imagination." When law students hear of a proposed legal rule for guiding – or judging – behavior, can they imagine situations in which the proposed rule would lead to unintended results or conflicts with deeply held norms? Can students use these hypothetical examples to critique the proposed rule and can they respond to such critiques? Legal interpretation with hypotheticals is not only an important form of legal argument, it epitomizes something fundamental about the nature of legal reasoning: attorneys and judges reason *about* legal rules, not just *with* them. A proposed rule can be seen as a hypothesis, a tentative assumption made in order to draw out and test its normative, logical or empirical consequences. A hypothetical is an imagined situation that helps to test such a hypothesis; it is used to help draw out the various types of consequences of a proposed rule. U.S. Supreme Court Justices are famous for posing hypotheticals during oral arguments to evaluate proposed rules for deciding a case. As a step toward getting students to develop skills at reasoning with hypotheticals, the LARGO ("Legal ARgument Graph Observer") intelligent tutoring system supports

them as they study U. S. Supreme Court argument transcripts in which expert reasoning with hypotheticals unfolds. LARGO enables students to represent these arguments in simple graphic terms and then to reflect on the arguments' merits and significance.

Researchers aiming to develop systems that engage students in argument or improve their argumentation skills have been drawn to graphical representations for a number of reasons. From a cognitive perspective, graphical representations can reduce the students' cognitive load and reify important relationships. Thus, it is hypothesized, they facilitate reasoning about texts and the acquisition of interpretive skills [1, 5]. While the use of two simultaneous representations can increase cognitive load, the complementary strengths of a textual and graphical argument form can better guide students in their analysis. Second, intelligent tutoring systems can provide feedback on graphical argument representations while finessing the fact that natural language processing remains difficult. A student-made graph provides the system with information about their thinking that, even if it does not rise to the level of complete understanding, can be leveraged to provide intelligent help [8, 9].

Unlike earlier systems, LARGO's diagrammatic language is geared toward the specific form of argumentation that it supports, as opposed to a general argument representation, enabling LARGO to provide more task-specific targeted feedback. We conducted a controlled experiment to test the hypothesis that LARGO's graphical representations and feedback help students learn better than with the help of a purely text-based tool. This paper presents the results of this system evaluation.

1. Example and Model of Legal Interpretation with Hypotheticals

We illustrate the process of legal interpretation with hypotheticals with an extract from the oral argument in the case of Kathy Keeton v. Hustler Magazine, 465 U.S. 770 (1984). This case deals with one of the first technical legal concepts that law students encounter in the first year: personal jurisdiction, a court's power to require that a person or corporation appear in court and defend against a lawsuit. These cases often pit the principle that a state may redress wrongs committed within the state against the U.S. Constitutional principle of "due process" (minimum procedural safeguards against the arbitrary exercise of government power), especially when a court sitting in one state asserts power over a nonresident of that state. The plaintiff, Kathy Keeton, sued Hustler Magazine, an Ohio corporation with its principle place of business in California, in U.S. District Court in New Hampshire. She claimed that Hustler had libeled her in five articles. She was not a resident of New Hampshire and had almost no ties there. Hustler's contacts with New Hampshire included the monthly sale of up to 15,000 magazine issues there. At the time, New Hampshire was the only state in which Ms. Keeton was not barred under a state statute of limitations from making her claim.

In U. S. Supreme Court oral arguments, each side gets one half hour to address the issue before the Court. The extract shown in Figure 1 illustrates key argument moves used during these sessions, modeled as in [4]. The left column labels the different argument elements, such as proposed tests, hypotheticals, and ways of responding to hypotheticals, while the right contains the actual argument text. "Q:" indicates a Justice's question. Mr. Grutman represents Ms. Keeton. He begins by proposing a rule-like test for deciding the problem in a manner favorable to his client (line 14). Such proposals often include supportive reasons, such as that the proposed test explains past case decisions or is consistent with, or best reconciles, principles and policies

underlying the law. Justices may respond by posing a hypothetical (lines 55, 57, 59), which may simply be a query about the test's meaning (line 55 & 57), or may underscore the test's overly broad scope (line 59). The advocate has to rebut or otherwise reply to the challenge to maintain his argument's credibility. He may attempt to justify his proposed test by arguing that the supposedly disanalogous counterexample (i.e., the hypothetical) is really analogous to the current fact situation (cfs), in effect disputing that the proposed rule should yield a different result when applied to the counterexample than when applied to the cfs (as in lines 56 & 58). Or, he may distinguish the hypothetical from the cfs (as in lines 64 & 66).

→ Proposed test of Mr. Grutman for Plaintiff Keeton	14. GRUTMAN: The synthesis of those cases holds that where you have purposeful conduct by a defendant directed at the forum in question and out of which conduct the cause of action arises or is generated that satisfies the formula of those minimum contacts which substantial justice and reasonable fair play make it suitable that a defendant should be hailed into that court and be amenable to suit in that jurisdiction.
← J.'s hypo	55. Q: Would it apply in Alaska?
→ Response: analogize cfs/hypo	56. GRUTMAN: It would apply, Mr. Justice Marshall, wherever the magazine was circulated. It would apply in Honolulu if the publication were circulated there. It would apply theoretically and, I think, correctly wherever the magazine was circulated, however many copies were circulated
← J.'s hypo	57. Q: Just to clarify the point, that would be even if the plaintiff was totally unknown in the jurisdiction before the magazine was circulated?
→ Response: analogize cfs/hypo	58. GRUTMAN: I think that is correct, Mr. Stevens, so long as Alaska or Hawaii adheres, I believe, to the uniform and universal determination that the tort of libel is perpetrated wherever a defamatory falsehood is circulated. Wherever a third person reads about it, there is that harm
← J.'s hypo	59. Q: What if the publisher had no intention of ever selling any magazines in New Hampshire
	60. GRUTMAN: A very different case, Mr. Justice White.
→ Response: distinguish cfs/hypo	64, 66. GRUTMAN: … because in that case you could not say, as you do here, that you have purposeful conduct. There you have to look for other -- I think your phrase is affiliating circumstances, other connections, judicially cognizable ties --

Figure 1. Examples of interpretive reasoning with hypotheticals in oral argument in Keeton v. Hustler.

2. The LARGO Intelligent Tutoring System

From the viewpoint of legal pedagogy, oral argument examples like that above are worth studying, but they are challenging materials to beginning law students. A program that engages students in reflecting upon such expert examples could help; it could bring the general argumentation principles to the forefront and at the same time require that students be active learners. LARGO allows law students to graphically represent the dialectical pattern of hypothetical reasoning. Figure 2 shows an example graph based upon the Keeton case. This graph was prepared by a naïve user for the purpose of illustration (since Keeton was part of the post-test for our study the subjects did not use LARGO with this case). The left side of the screen shows parts of the oral argument transcript. Below that is an advice button and a palette of the basic graphical elements for markup. These elements include nodes representing proposed tests, hypotheticals, the cfs, and relations (like modification, distinction and analogy). Graphs are constructed by dragging these elements into the workspace and filling in appropriate text. Students can also link graph elements to parts of the transcript using a highlighting feature (e.g., the yellow node in Fig. 2 is linked to line 58 of the transcript).

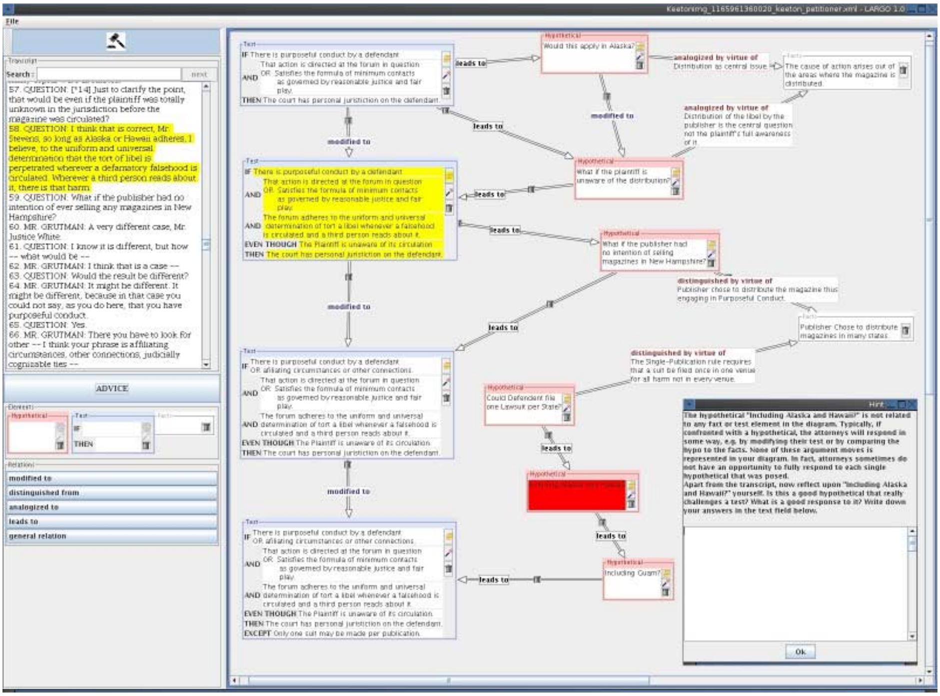


Figure 2. LARGO Representation of Keeton Case Oral Argument

When a student clicks on the Advice button, LARGO provides advice on improving their diagram. It detects three categories of diagram characteristics (a characteristic can either be a potential diagram weakness [9] or an opportunity for reflection). *Structural* characteristics involve portions of the graph where the relations among elements either do not correspond to the model of interpretive reasoning illustrated in section 1 (weaknesses) or correspond to an interesting or unusual constellation (opportunities for reflection). For example, hypotheticals are typically related directly to a test or other advocate response during oral argument. Novices often fail to represent these relations with diagram links. LARGO is capable of detecting this kind of diagram characteristic, and considers it a weakness. In Figure 2 LARGO provides a structural hint in the dialog box at the bottom right, pointing out that the hypothetical “What about Alaska and Hawaii” is not connected to any test element. *Context* weaknesses involve situations where the student’s graph does not contain elements corresponding to key passages in the argument transcript that have been marked up in advance by a law professor. For example, the student may have missed a proposed test, or may have represented a test as a hypothetical. Finally, *content* weaknesses are poor formulations of proposed tests identified in the transcript.

Internally, LARGO relies on two analysis mechanisms: First, a graph-grammar formalism is used to detect context and structural weaknesses or opportunities for reflection. Second, a collaborative filtering technique is used to remediate content weaknesses. When students, after reading an attorney’s proposed test in the transcript, enter a formulation of that test into their diagram, they are prompted to rate their formulation of that test against others produced by their peers. This information enables LARGO to derive a ranking of the formulations of a given test by all students [9].

Given the ill-defined nature the legal domain [6], one cannot always be certain that a diagnosed graph weakness represents an inaccurate rendition of the transcript, or how it should be "repaired". It may be that the particular line of argument is unusual (it is difficult to foresee all such possibilities). Or the Justices may have abandoned a line of questioning before a standard argument pattern could be completed. Therefore, LARGO's feedback is typically couched as invitations to reflect or as self-explanation prompts. These types of prompts have proven effective as a metacognitive strategy also in ill-defined domains [12]. For example the hint in Figure 2 prompts the student to think about the fact that one of the hypotheticals, highlighted red, is unconnected to any test or fact. If that was indeed the case in the transcript, then the diagram should reflect that (and thus is fine as it is), but if not the advice may lead the student to repair it. In either case it seems useful to invite the student to reflect on the role of this hypothetical.

3. Experimental Procedure

We conducted an experiment comparing LARGO's graphical representations and advice with a text-based alternative which simulates the process of examining the text with a notepad by allowing students to highlight portions of the text and enter notes in a text pane (without feedback). Our hypothesis was that the graphical format and advice would help students better identify and understand the argument components. The experiment was conducted in concert with the first-year Legal Process course at the University of Pittsburgh and with the professors' permission. The cases examined in the study centered on personal jurisdiction which was part of their coursework. We invited students to volunteer for the study, they received \$80 for their participation. Students were assigned randomly to the conditions, balanced in terms of LSAT (Law School Admissions Test) scores. 38 students began the study, 28 completed it.

The experiment involved four sessions of 2 hours each over a four-week period. The first was a pre-test and an introduction to the software. In the second and third sessions, the students worked with extracts of the oral arguments from two personal jurisdiction cases. In the Experimental condition, students represented them graphically using LARGO with the help of the feedback mechanisms. In the Control condition, students were instructed to highlight relevant passages and take notes using the notepad. Session four consisted of the post-test. The pre- and post-tests, designed to assess students' argument and hypothetical reasoning skills, comprised five types of multiple-choice questions: 1) Legal argument-related questions of a type considered for inclusion in the LSAT [11]; 2) General questions about the use of hypotheticals in legal argument; 3) Questions that explored the use of hypotheticals for argument in a non-legal, intuitive domain about the policies of a tennis club; 4) Near-transfer argumentation questions about tests, hypotheticals, and responses in a new personal jurisdiction case (Keeton); 5) Far-Transfer argumentation questions drawn from a legal domain (copyright law) with which students were not likely to be familiar. The first three types appeared on both pre- and post-test; the last two only on the post-test.

4. Results and Discussion

We first computed a single overall pre-test and post-test score for each subject, which included all survey items with multiple choice answers. We also computed subscores

for each of the five specific question types described above. No significant difference on overall or subscores existed between the two conditions on the pre-test. The post-test scores for the Experimental subjects were consistently higher than the Control subjects' scores both with respect to overall scores and subscores. However, these differences were not statistically significant ($t(1,26)=.92$, $p>.1$, for overall scores). For some question types, the effect size between conditions would have been considerable if the results had reached the level of significance ($t(1,26)=1.22$, Cohen's $d=.60$, $p>.1$, near transfer problem; $t(1,26)=1.25$, $d=.45$, $p>.1$, "Tennis Club" post-test questions).

We then divided up students by LSAT score, creating a "Low" group containing 10 students, a "Med" group with 9, and a "High" group with 8 (the group sizes vary slightly to ensure that all students with the same score are in the same group; all students in the Med group had the same LSAT score). One student did not have an LSAT score and was not included. The results of these three groups differed considerably ($F(2,25)=4.15$, $p<.05$, for the overall score, similar results for most subscores). The students in the Low group (average post-test score .54) scored significantly lower than those in the High group who had an average of .63 ($p<.01$), consistent with the predictive value claimed for the LSAT scores. The Med group was in between the two (average .55). In a subsequent analysis we found that for the students in the Low group, there was a condition effect for the near-transfer questions (Keeton case). We hypothesized that lower aptitude Experimental subjects would do better than the lower aptitude Control subjects. A 1-sided t-test showed an effect size of 1.99 ($p<.05$). There were no pre-test differences between these groups ($p>.8$).

In order to determine whether LARGO helped students understand some parts of the argument model better than others, we classified the questions in the pre- and post-tests in terms of which aspect of the argument model they relate to most: tests, hypotheticals, relations between the two, responses to hypotheticals, or none (general questions). This post-hoc analysis revealed another effect: students in the Low and Med groups benefited from LARGO and did better on post-test questions that ask them to evaluate a hypothetical with respect to a given test. For the Low group and the combined Low+Med group, the difference was significant (Low+Med: $t(1,17)=2.73$, $d=1.00$, $p<.05$, 1-sided), but not for the whole group (Low+Med+High). The differences between conditions for other LSAT groups and test items were not significant. Below is an example of a "hypothetical evaluation with respect to test" question where the Experimental subjects outperformed the Control subjects:

> Assume that Mr. Grutman's proposed test is as follows: If … defendant has engaged in purposeful conduct …, and … satisfies the minimum contacts … (see Figure 1, line 14).
>
> The following hypothetical was posed in the oral argument. It is followed by four explanations why the hypothetical is or is not problematic for Mr. Grutman's proposed test. Please check ALL of the explanations that are plausible.
>
> "… if the plaintiff was totally unknown in the jurisdiction before the magazine was circulated?" (see Figure 1, line 57)
>
> o The hypothetical is problematic for Mr. Grutman's proposed test. The decision rule applies by its terms, but arguably the publisher should not be subject to personal jurisdiction in the state under those circumstances.
> o The hypothetical is not problematic for Mr. Grutman's proposed test. The decision rule applies by its terms, and the publisher should be subject to personal jurisdiction in the state under those circumstances.
> o …
> o …

These results support our research hypothesis (although perhaps fall somewhat short of decisively confirming it). For the Low group, the use of LARGO with its graphical argument representation and feedback in the form of tailored self-explanation prompts led to significantly better learning of argumentation skills than the use of traditional note-taking techniques, as measured in a near transfer problem which involved argumentation questions about a novel case in the same legal domain as those studied in the experiment. For the far transfer problem (a novel case in different legal domain) this effect was not found. We are also intrigued by the finding the Low+Med Experimental subjects apparently learned more about evaluating hypotheticals with respect to tests (not restricted to near transfer questions) than their Control counterparts. As the example illustrates, this skill is central to what LARGO is designed to teach: The relationship between tests and hypotheticals is the essence of oral argument.

One important question is why a significant difference was found on this particular question type and not on the other argumentation-model-related items (tests, hypotheticals, responses to hypotheticals). One possible explanation is that LARGO's graphical language distills the essence of the oral argument visually, explicitly identifying the relations between tests and hypotheticals (cf. Figure 2). Our data suggests that less skilled students benefited from creating and reflecting on these diagrams (with the help of LARGO's feedback), whereas more skilled students may have been able to understand the complex relations without aid. One can argue that for the other argumentation-model-related items, the specific graph structure or advice features that LARGO employs are not sufficient to differentiate it from purely text-based annotation tools. The student's ability to formulate a good test might not be supported to a great extent by a graphical representation format or prompts LARGO offers. However, as the near-transfer effect for the Low group shows, the less skilled students do benefit from LARGO also on a general level.

In summary, our analysis of the study results so far shows that the LARGO ITS is a valuable tool for those learners who do not (yet) have the ability to learn argumentation skills from independent study of argument transcripts. This group seems to benefit from the scaffold that the diagrams and the feedback offer. For the more advanced/skilled students, LARGO did not prove to be significantly better (but also not worse) than traditional learning resources such as a notepad and a highlighter. Yet, an analysis of the log files indicates that this group too appreciated LARGO's feedback and advice functions. On average (across all sessions of the study and all students in the Experimental condition), the advice button was pressed 10.1 times per transcript (i.e., per hour). All 3 groups frequently requested advice: Low, 12.3; Med 6.2; High 17.9. In 75% of these cases, students selected one of the three short hint titles that LARGO presented in response to their hint request, and read through the detailed feedback related to the selected hint title. The use of the advice did not decrease over time. In the later sessions, the average number of help requests was even higher than in the earlier sessions (12.2 and 8.6 in the last two transcripts vs. 7.3 and 9.8 in the first two), evidence that the students must have considered the advice to be valuable.

5. Conclusions and Outlook

A study carried out in a first-year law school course provides support for the hypothesis that a diagrammatic language, combined with feedback that points out weaknesses and opportunities for reflection in students' argument diagrams, helps students learn to

apply a general model of hypothetical reasoning, as they study transcripts of arguments made by highly-skilled experts. For lower-aptitude students, use of LARGO's diagramming and feedback functions was more effective than traditional note-taking techniques. Specifically, within this group, those who used LARGO learned better to analyze new argument transcripts in the same area of the law, even when they studied the new transcript without the use of LARGO. They also learned better to reason about how a hypothetical might relate to a proposed test, a key element of hypothetical reasoning. The study thus demonstrates the benefits and potential of graphical representations in intelligent tutoring systems for argumentation, especially with lower-tier students. In an earlier study [3], we found benefits of (non-dynamic) self-explanation prompts based on the same general model of hypothetical reasoning on which LARGO is based. This study was carried out in the context of a program to help disadvantaged students prepare for law school. Taken together, these studies suggest that LARGO would be a valuable addition to regular law-school curricula and to preparatory programs for disadvantaged students.

At present we are continuing the analysis of the study data. We will conduct further analyses of students' log files, argument graphs, and text notes. We are particularly interested in determining whether the Experimental subjects were better able to find and relate key tests and hypotheticals than their Control counterparts. We will also determine how the variability of students' argument graphs speaks to the ill-defined nature of the domain [6]. Furthermore, we will investigate which of LARGO's hints were most helpful. LARGO represents one of the first full-scale implementations of the collaborative filtering idea in an educational application. We will verify whether the filtering did, in fact, accurately assess students' formulations of the tests they identified in the argument transcripts.

References

[1] Ainsworth, S. (1999). The functions of multiple representations. *Computers and Education 33,* 131-152.

[2] Aleven, V. (2006). An intelligent learning environment for case-based argumentation. *Technology, Instruction, Cognition, and Learning.*

[3] Aleven V., Ashley, K.D., & Lynch, C., (2005). Helping Law Students to Understand US Supreme Court Oral Arguments: A Planned Experiment. Proc. of 10th Int. Conf. on AI and Law. Bologna Italy.

[4] Ashley, K. (2007). Interpretive Reasoning with Hypothetical Cases. In Proc. 20th Int'l FLAIRS Conference, Special Track on Case-Based Reasoning, Key West, May.

[5] Larkin, J. & Simon, H. A. (1987). Why a diagram is (sometimes) worth ten thousand words. *Cognitive Science, 11,* 65-99.

[6] Lynch, C., Ashley, K, Aleven, V., & Pinkwart, N. (2006). Defining Ill-Defined Domains; A literature survey. In V. Aleven, K. Ashley, C. Lynch, & N. Pinkwart (Eds.), Proc. Workshop on Intelligent Tutoring Systems for Ill-Defined Domains at the 8th Int'l Conf. on Intelligent Tutoring Systems, 1-10.

[7] Muntjewerff, A. J., & Breuker, J. A. (2001). Evaluating PROSA, a system to train solving legal cases. In J. D. Moore et al. (Eds.), *Proc. of AIED 2001* (pp. 278-285). Amsterdam: IOS Press.

[8] Paolucci, M., Suthers, D., & Weiner, A. (1996). Automated advice-giving strategies for scientific inquiry. In C. Frasson, G. Gauthier, & A. Lesgold (Eds.), Proc. ITS '96 (pp. 372-381). Berlin: Springer.

[9] Pinkwart, N., Aleven, V., Ashley, K., & Lynch, C. (2006). Toward Legal Argument Instruction with Graph Grammars and Collaborative Filtering Techniques. In M. Ikeda, K. Ashley, & T. W. Chan (Eds.), Proceedings of ITS 2006 (p. 227-236). Berlin: Springer.

[10] Ravenscroft, A. & Pilkington, R.M. (2000) Investigation by Design: developing dialogue models to support reasoning and conceptual change. *Int. Journal of AI in Education, 11*(1), 273-298.

[11] Reese, L. & Cotter, R. (1994) A Compendium of LSAT and LSAC-Sponsored Item Types 1948-1994. Law School Admission Council Research Rept. 94-01.

[12] Schworm, S., & Renkl, A. (2002). Learning by solved example problems: Instructional explanations reduce self explanation activity. In Proc. Ann. Conf. of the Cogn. Sci. Society (pp. 816-821). Erlbaum.

Motivation

This page intentionally left blank

Relating Machine Estimates of Students' Learning Goals to Learning Outcomes: A DBN Approach

Carole R. BEAL & Lei QU
Information Sciences Institute - USC
4676 Admiralty Way, Marina del Rey CA 90292 USA
cbeal@isi.edu

Abstract. Students' actions while working with a tuoring system were used to generate estimates of learning goals, specifically, the goal of learning by using multimedia help resources, and the goal of learning through independent problem solving. A Dynamic Bayesian Network (DBN) model was trained with interface action and inter-action interval latency data from 115 high school students, and then tested with action data from an independent sample of 135 students. Estimates of learning goals generated by the model predicted student performance on a post-test of math achievement, whereas pre-test performance did not.

Introduction

There is growing interest in modeling students' engagement with tutoring systems, meaning the transient shifts in attention, cognitive effort and emotions associated with learning situations. Specific behaviors associated with engagement have been identified, including facial expressions, changes in posture, and pressure of the hands on the keyboard, among others sources of information that can be detected by machine tutors [1, 2, 3]. However, at present, few schools are equipped with gaze-tracking cameras or devices to sense student posture or pressure grip, at least not in numbers sufficient for classes that may include 35 or more students in a computer lab. Talk-aloud procedures can yield rich data to assist with system development and evaluation, but would be disruptive in crowded in-vivo classroom conditions. Thus, our focus in the present study was on non-intrusive alternatives to building rich learner models from students' interactions with the ITS interface, including actions and inter-action intervals – data that can be captured automatically as students work [4].

The study builds on our prior work with processing students' actions as they worked with a mathematics tutoring system, and classifying the student's behavior on a math problem, i.e., whether the student had guessed, solved the problem without errors or using multimedia help, abused help features, or viewed help to understand the solution path [5]. These action patterns were strongly related to learner motivation, meaning the constellation of domain-specific beliefs associated with achievement, including self-concept, domain value, and related constructs. The proportions of different action patterns for individual students were also related to their achievement in math, as rated by their teachers.

The present study extends the previous work in two ways: First, considerable

work has focused on students' use or failure to use ITS help features effectively [6]. Yet relatively little attention has been directed to students who are engaged with the ITS but are not interested in utilizing help features, even though theories of self-regulated learning suggest that, for some students, self-directed practice and error correction may be as effective as using ITS help features [7, 8, 9, 10]. Thus, our goal was to create a model that would recognize both learning strategies. The second goal was to relate estimates of the learner's goals to a measure of learning from the ITS activity: here, performance on a post-test of math problem solving. If the learning goal estimates generated by the model are reasonably accurate, then they should relate to how much students actually learn from the ITS, and to which students show significant improvements from pre- to post-test.

1. Modeling learners' goals

1.1 Dynamic Bayesian Network models

Recognizing student learning goals is challenging even for human tutors, because observable actions such as staring at the screen can be ambiguous; in addition, the learner's goals may change from moment to moment. In order to handle this diagnostic and temporal uncertainty, our model uses a Dynamic Bayesian Network (DBN) [11] to estimate students' goals of learning. The Dynamic Bayesian network is a graphical model that encodes probabilistic relationships among state variables [12]. The relationship between any set of state variables can be specified by a joint probability distribution; researchers have recently extended such models into dynamic situations that evolve over time, including inferences about cognitive and affective states [13].

1.2 Data source and model training

The DBN model was initially constructed using a dataset from high school students (N = 115) who worked with Wayang Outpost, a mathematics tutoring system. Each problem in the Wayang Outpost tutoring system included five answer options. Students could choose an answer at any point and receive feedback (e.g., when an answer was given by student selecting an option, the feedback with a red "X" indicated the answer was wrong, whereas the feedback with a green checkmark indicated the answer was right). Students could also request a multimedia explanation of the solution by clicking the "Hints" icon. Explanations were constructed as an ordered sequence of individual hints leading to the correct answer. For example, angles that are important for the solution are highlighted with an animation. Individual hints included information presented in one modality (e.g., text, or animation, or audio) to avoid excessive cognitive load, but the complete explanation for a problem included hints with a range of modalities. After students finished a problem, they could click the "Next" icon to proceed to a new problem.

Students completed 2,154 problems in Wayang Outpost, and 6,468 interface actions were logged into the database with a timestamp. The actions included:

- Begin: Student starts a new math problem
- Hint: Student clicks icon to request a multimedia hint
- Correct: Student selects the correct answer
- Attempt: Student selects an incorrect answer

Each of the actions was linked to a data record < *Student, ProblemID, ActionType, Timestamp* >, and to a calculation of the time the student spent on the action < *TimeAction* >. These action records were used to build the Conditional Probability Tables (CPTs) for the DBN model; the model nodes are shown in Figure 2.

Estimates for two learning goals were generated by using the next action of the student, and an evaluation of *TimeAction* against temporal thresholds. Threshold values were established in prior work on the basis of latencies associated with proficient students, e.g., if a high-achieving student required 10 or more seconds to read the problem before responding, a student who chooses an answer in less time was unlikely to have read the problem and may have been guessing. Estimates were updated after each student action, and reset after each problem was completed.

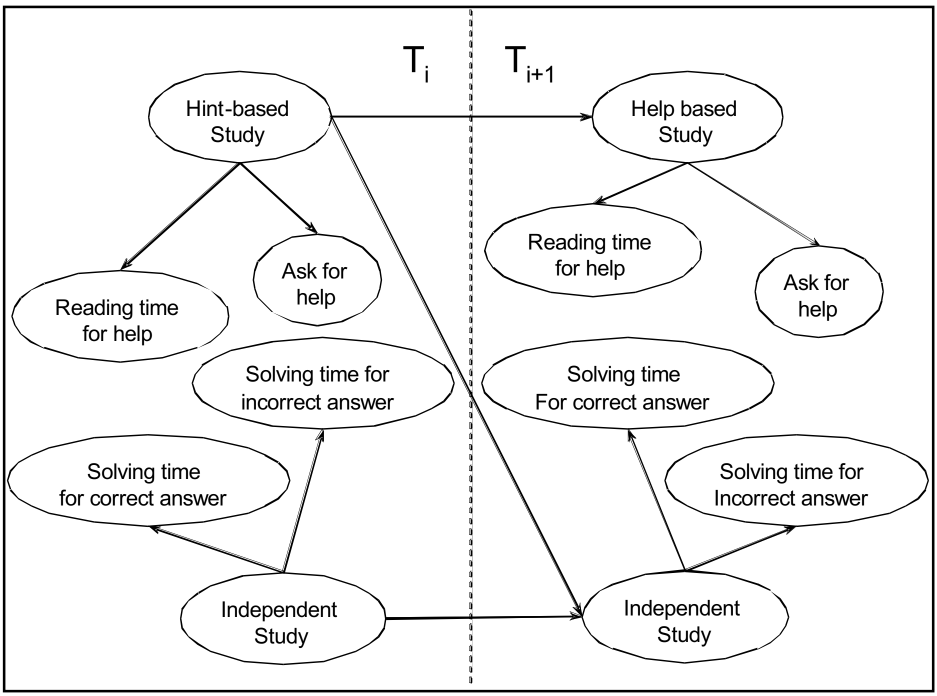


Figure 2. DBN model with estimates for Hint-based Study and Independent Study

Hint-based Study represented situations where the student read the problem (using latencies of 10+ seconds), and solved the problem using multimedia help. Hint-based study was defined in the model to have four states: High, Average, Low, and None (None because the student might not request hints at all on a problem). Independent Study represented situations where the student appeared to read the problem, did not request help, and attempted to find the solution alone. These situations included cases where the student entered the correct answer as well as cases where the correct

answer was preceded by incorrect answers, but latencies suggested that he or she was not attempting to game the system. Four states were defined: Very high, High, Average, or Low (Low because as long as the problem was available it was at least possible the student was trying to solve it, even if her or his behavior suggested otherwise).

2. Results

2.1 Data source

Data from an additional 135 high school students in a different school system were used to evaluate the DBN model. These students first completed a pre-test of math skills, worked with Wayang Outpost during their mathematics classes, and then completed a post-test. There were two versions of the test, counter-balanced across students. Tests were completed at the computer and scored automatically. Quartile scores were used to classify students into pre- and post-test groups (High, Average, Low, Very Low math proficiency).

Students' actions with the tutoring system were logged and time-stamped as described above. When each math problem was completed, the model's estimates for Hint-based Study (ranging from 0-3) and Independent Study (1-4) were summed and divided by the total time for the problem. These values were then averaged for the number of problems completed by the student. Quartile scores were then used to classify students into four groups: students with High, Average, Low or No Hint-Based Study scores, and students with Very High, High, Average, or Low Independent Study scores.

2.2 Relation of prior math achievement to Independent Study estimates

It seemed reasonable to predict that students with better initial math skills would have higher Independent Study estimates: such students should have the skills required to find the problem solutions on their own without needing to view the multimedia help features. Consistent with this prediction, there was a significant relation between Pre-test score and Independent Study scores, $F(1,33) = 92.099$, $p < .001$, with $r^2 = 0.409$. A one-way Analysis of Variance with Independent Study Group as the grouping factor and Pre-test score as the dependent measure indicated a significant effect of Independent Study group, $F(3,131) = 41.549$, $p < .001$. Tukey's HSD tests with alpha levels set to 0.05 indicated that Very High Independent Study students had significantly higher scores on the math pre-test than other students.

2.3 Relation of DBN estimates and learning outcomes

We were also interested in whether the model's estimates of ITS learning goals were related to students' performance on our outcome measure of learning: scores on the math post-test. For example, a student with low Independent Study and low Hint-based Study scores would not be expected to improve as the result of ITS tutoring because the model suggests that the student was not engaged with the ITS. In contrast, it would be reasonable to predict that a student who had a high Hint-based Study score might show improvement on the post-test.

To investigate the relation of actions with the ITS to learning outcomes, we

assigned students to Learning Goal groups based on the cross-tabulation of Hint-based Study and Independent Study group membership. Table 1 shows these data. For example, there were 14 students who had relatively low estimates for both Independent and High-based Study (Group 0). Students who were in the High Hint-based Study but the Low Independent Study groups were assigned to Group 1 (N = 53). Those with low Hint-based Study but High Independent Study estimates were in Group 2 (N = 53). Students who were High for both measures were in Group 3 (N = 15). This could occur if, for example, a student read the problem, attempted to solve it, and then viewed the help features.

Table 1. Cross-tab of Hint-based Study and Independent-Study groups

	No & Low Hint-based Study Groups	Medium & High Hint-based Study Groups
Low & Medium Independent Study Groups	N = 14 Group 0 (LOW)	N = 53 HBS > IS (Group 1)
High and Very High Independent Study Groups	N = 53 IS > HBS (Group 2)	N = 15 HIGH (Group 3)

To evaluate the relation with learning outcomes, a logistic regression was conducted to test the contributions of Pre-test Group (High, Average, Low or Very Low Math Achievement), Learning Goal Group (Low, HS>IS, IS>HS, High) and the interaction term to predictions of Post-test Group membership. The results indicated that the whole model was significant, $\chi(7) = 56.756$, $p < .001$. Pre-test Group did not contribute significantly to the model, $\chi\ (1) = 0.367$, N.S. This indicates that knowing how the student performed on the pre-test of math skill was not sufficient to predict her or his post-test score reliably. However, there was a significant contribution of Learning Goal Group to the model, $\chi(3) = 16.928$, $p < .001$. This effect indicates that students' learning goals, as estimated by the DBN model, accounted for significant variance in their scores on the Post-test.

The main effect of Learning Goal was qualified by a significant interaction between Pre-test Group and Learning Goal Group terms in the logistic regression model, $\chi(3) = 14.866$, $p < .001$. Mean pre- and post-test scores for the four Learning Goal Groups are shown in Figure 3. As may be observed in the Figure, students whose actions indicated both low use of hints and also low effort to solve problems independently (Group 0) had low scores on the pre-test, and they did not improve much as the result of the tutoring activity. In contrast, students with similarly low pre-test scores but relatively high Hint-Based Study Estimates (Group 1) showed significant improvement on the post-test. The picture is different for students with high Independent Study scores (Group 2). These students had better math skills to begin with (indicated by their higher pre-test scores); they were more likely to work independently on the math problems; and their independent study was associated with improvement on the post-test.

3. Discussion

In the present study, time-stamped student interface actions were used to train a DBN model to generate estimates of Hint-based Study and Independent Study: two pathways to learning suggested by theories of self-regulated learning. The trained DBN model was then used to generate learning goal estimates for a second independent sample. Students varied in their relative rates of Hint-based Study and Independent Study: Some did not use the help features very much if at all, whereas others did so at a high rate. Some students who did not use multimedia help appeared to work hard to figure out the solutions on their own, whereas others simply guessed their way through the problem to the correct answer. Self-regulation theories would predict that these behaviors should affect how much students learn [7, 8, 14]. Consistent with this prediction, students with low estimates of both learning goals did not show significant improvement in their test performance, whereas others with high Hint-based Study or high Independent Study estimates did improve. Thus, the DBN estimates were related to an independent measure of student learning.

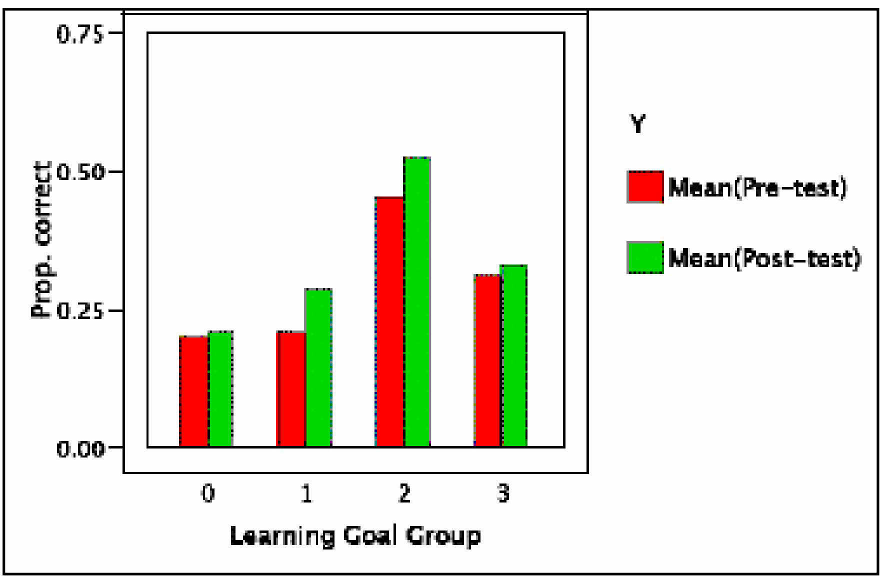


Figure 3: Mean scores pre- and post-test scores for Learning Goal Groups: Group 0 is Low Engagement, Group 1 is High Hint-based Learning, Group 2 is High Independent Study, Group 3 is High Hint and Independent Study

We also found that students' initial math achievement was related to their learning goals, as estimated by the DBN model. Specifically, students who started the ITS activity with weak skills, as indicated by low pre-test scores, tended to have high Hint-based Study scores relative to students with better math skills who were more likely to have high Independent Study scores. Perhaps this is not surprising: students who already know at least some of the mathematics concepts and skills targeted in the

ITS would have a better chance to solve the problems through effort. What is more interesting is that many of the better students chose to learn through their own efforts rather than by viewing the ITS help features. Although we did not have the opportunity to ask students why they preferred to work on their own, a few informal observations along with other work suggests that attributions may play a role. Finding the answer on your own, even if you make many errors, can be interpreted as support for your ability, whereas using multimedia help may undermine this interpretation. Interestingly, some students did view the multimedia help *after* they had arrived at the correct answer, saying that they wanted to confirm that they had done the problem correctly.

The results also indicated, for students with poor math skills, use of ITS help was associated with improvement from pre- to post-test. However, there were some students with poor math skills who did not use the hints and did not learn much, if anything, from the ITS activity. There were relatively few such students (10% of the sample). However, it will be important in future work to learn more about why these students were reluctant to engage in hint-based study, and how to increase their engagement with the ITS. The DBN model offers a way to recognize these cases from students' actions, as a first step towards effective intervention.

One question is why Group 3 students, whose actions suggested the highest level of engagement (both Hint-based and Independent Study estimate), did not show significant improvement from pre- to post-test. If, as estimated by the DBN model, these students were highly engaged in learning from the ITS, why did they not perform better on the post-test? One possibility is that these students' actions were not classified correctly by the DBN, which was sensitive to the latencies associated with students' actions. For example, if the student takes some time to review the problem before attempting an answer, the model will assign higher values for Independent Study than when the student attempts to answer very quickly. However, it is quite possible that a long latency before the first response might reflect student boredom rather than engagement with problem solving. In addition, at least some correct answer selections that would increase Independent Study estimates in the model might have been due to lucky guesses; there are five answer options for each math problem, so chance performance would be 20%. Thus, Group 3 students' temporal data may have suggested high engagement when the students were actually bored or detached. Even so, it should be noted that there were only 15 students in this group, meaning that the DBN model generated estimates of Hint-based Study and Independent Study that were strongly related to the outcome measure (post-test scores) for 89% of the sample. Additional work will be required to identify why the model did not function well for the remaining students.

4. Future work

In the present study, students' actions with the ITS interface were used to infer learning goals through a Dynamic Bayesian Network. Although support for the model was provided by the link to learning outcomes, it will be important in future work to validate the model with real-time data about cognitive workload and attention. We also plan to include a measure of self-regulation, to learn if students' strategic behavior in classroom-based learning predicts their Hint-based Study and Independent Study estimates while working with an ITS [14].

Acknowledgments

The Wayang Outpost project has been supported by National Science Foundation grants HRD 0120809, HRD 0411532, and REC 0411886. The views expressed are not necessarily those of the funding agency. We would like to thank the principals and teachers in the Pasadena Unified School District and the Bravo Medical Magnet High School in the Los Angeles Unified School District for their support of the research. We would also like to thank project members Erin Shaw, Jean-Philippe Steinmetz, Hyokyeong Lee, Mike Birch, and Sharan Volin for their assistance with the research.

References

[1] D'Mello, S. K., Craig, S. D., Gholson, B., Franklin, S., Picard, R., & Graesser, A. C. (2004). Integrating affect sensors into an intelligent tutoring system. In Affective interactions: The computer in the affective loop. Proceedings of the 2005 International Conference on Intelligent User Interfaces, pp. 7-13. New York: AMC Press.

[2] Qu, L., & Johnson, W. L. (2005). Detecting the learner's motivational states in an interactive learning environment. In C-K. Looi et al. (Eds.), Artificial intelligence in education: Supporting learning through intelligent and socially informed technology, pp. 547-554. Amsterdam: IOS Press.

[3] Vicente, A., & Pain, H. (2002). Informing the detection of the student's motivational state: An empirical study. In 6th International Conference on Intelligent Tutoring Systems, pp. 933-943. Biarritz, France.

[4] Beck, J. E. (2005). Engagement tracing: Using response times to model student disengagement. In C-K. Looi et al. (Eds.), Artificial intelligence in education: Supporting learning through intelligent and socially informed technology, pp. 88-95. Amsterdam: IOS Press.

[5] Beal, C. R., Qu, L., & Lee, H. (2006). Classifying learner engagement through integration of multiple data sources. Proceedings of the American Association of Artificial Intelligence annual meeting, Boston MA.

[6] Baker, R. S., Corbett, A. T., Koedinger, K. R., & Roll, I. (2005). Detecting when students game the system, across tutor subjects and classroom cohorts. In L. Ardissono, P. Brna, & A. Mitrovic (Eds.), User Modeling, pp. 220-224. Berlin: Springer-Verlag.

[7] Karabenick, S. (1998). Strategic help seeking: Implications for learning and teaching. Mahwah NJ: Erlbaum.

[8] Newman, R. S. How self-regulated learners cope with academic difficulty: The role of adaptive help seeking. In S. Pape, B. Zimmerman, B., & F. Pajares (Eds.), Theory into practice: Special issue: Becoming a self-regulated learner, pp. 132-138. Columbus, OH: The Ohio State University, 2002.

[9] Ryan, A. M., Patrick, H., Shim, S-O. (2005). Differential profiles of students identified by their teacher as having avoidant, appropriate, or dependent help-seeking tendencies in the classroom. Journal of Educational Psychology, 97, 275-285.

[10] Turner, J. C., Midgley, C., Meyer, D. K., Dheen, K., Anderman, E. M., Kang, Y, & Patrick, H. (2003). The classroom environment and students' reports of avoidance strategies in mathematics: A multimethod study. Journal of Educational Psychology, 95, 806-812.

[11] Dean, T., & Kanazawa, K. (1989). A model for reasoning about persistence and causation. Computational Intelligence 5(3):142-150.

[12] Nicholson, A. E., & Brady, J. M. (1994). Dynamic belief networks for discrete monitoring. IEEE Transactions on Systems, Man and Cybernetics, 24(11):1593-1610.

[13] Zhou X., & Conati C. (2003). Inferring user goals from personality and behavior in a causal model of user affect. Proceedings of IUI 2003, International Conference on Intelligent User Interfaces, Miami, FL, U.S.A. p. 211-218.

[14] Zimmerman, B., & Martinez-Pons, M. (1986). Development of a structured interview for assessing student use of self regulated learning strategies. American Educational Research Journal, 223, 614-62.

R. Luckin et al. (Eds.)
IOS Press, 2007

MathGirls: Toward Developing Girls' Positive Attitude and Self-Efficacy through Pedagogical Agents

Yanghee KIM[1], Quan WEI[1], Beijie XU[1], Youngah KO[1] and Vessela ILIEVA[2]
[1]*Department of Instructional Technology*
[2] *Department of Curriculum and Instruction*
Utah State University, USA
yanghee.kim@usu.edu
http://www.create.usu.edu

Abstract. MathGirls is a pedagogical-agent-based environment designed for high-school girls learning introductory algebra. Since females are in general more interested in interactive computing and more positive about the social presence of pedagogical agents, the environment provides a girl-friendly social learning environment, where pedagogical agents encourage the girls to build constructive views of learning math. This study investigated the impact of agent presence on changes in the girls' math attitude, their math self-efficacy, and their learning; on the girls' choice of their agents; and, on their perceptions of agent affability. The results revealed that the girls with an agent developed a more positive attitude and increased self-efficacy significantly, compared to the girls without an agent. Both groups showed significant increase in their learning. The study also showed that the girls chose female agents significantly more than male agents as their learning partners and perceived peer-like agents as more affable than teacher-like agents. The finding confirms the instructional value of pedagogical agents functioning as a social cognitive tool [1].

Keywords. Pedagogical agents, Math education, Attitudes, Self-efficacy

Introduction

Gender differences in academic interest and cognitive and interaction style are well documented [2-5]. In particular, the gender differences in motivation to learn science, technology, engineering, and math (STEM) have become more salient with recent concern about workforce imbalances in the fields of science and engineering. Many girls tend to hold beliefs, attributable mainly to social and cultural influences, that interfere with their learning of STEM and limit their pursuit of careers in those fields. Family, schools, and media are likely to impose stereotypic role expectations on girls. So girls need to be exposed to social environments that will encourage them to overcome ungrounded social stereotypes and to build constructive views of their competency in STEM. For women successful in mathematics-related careers, for instance, social influences such as encouragement and support from family members and teachers were

found to be the foundation on which those women built their academic confidence to overcome obstacles in their progression through male-dominant academic programs [6]. The stereotypic views of family, teachers, or friends seem to be societal issues unlikely to improve immediately. However, girl-friendly virtual social environments might be instantiated to counter undesirable social influences in the real world.

Studies of interactive computing have indicated that male and female learners show distinctive patterns in their perceptions and their performances based on design or functional characteristics of the environment. For instance, girls performed better with highly interactive hints, whereas boys performed better with non-interactive and low-intrusive hints [7]. Girls were more sensitive than boys to help messages [2]. Also, regarding multimedia interfaces, girls' priority was display, such as color and appearance, whereas boys' priority was navigational support and control [8]. College male and female students reacted to pedagogical agents socially, with the females perceiving them more positively than males [9]. Given these findings, the authors assumed that pedagogical agents--virtual digital peers or instructors--might create a rich social context favoring female students, by presenting a more interactive environment.

Such a view has ample support. Researchers in agent technology are shifting their view of agent systems from tools to actors [10], finding in their studies that socially intelligent anthropomorphized agents play a social role and thus can build social relationships with users [11]. Reflected in this conference theme, social motivational factors to promote learning have drawn attention of researchers in advanced technology for learning. This project, funded by NSF (#05226343, http://www.create.usu.edu), sought to create a girl-friendly virtual social environment using pedagogical agents to enhance high school girls' motivation toward learning math. In this environment, named *MathGirls*, pedagogical agents demonstrated advanced performances, encouraged the girls to be confident in learning math, and provided persuasive messages to help the girls build a positive attitude towards math learning. While the girls practiced solving algebra problems, the agents, 3-D images representing peers and teachers, proactively provided the girls with verbal persuasion as well as with cognitive guidance according to their performances.

This study was conducted to examine the impact of the MathGirls environment, whether the pedagogical agents in the environment influenced high school girls to change their math-related attitude and self-efficacy beliefs. Because attribute similarities between a social model and a learner, such as gender, age, and competency, often have been predictive for the learner's efficacy beliefs and achievements in traditional classrooms [12, 13] and because several studies have found that both male and female college students show differing perceptions and preferences according to agent characteristics [9, 11, 14, 15], the current study implemented four agents differing by gender and age, allowed the learners to choose their agents at the beginning of the lesson, and investigated their choice patterns. Detailed descriptions of the study follow.

1. MathGirls Environment

1.1 Curriculum

"Fundamentals in Algebra" was chosen as the curriculum, for two reasons. First, in the collaborating school district, ninth graders must take introductory algebra (Algebra I or Applied Algebra), regardless of their interests. This meant that the ninth grade sample

would represent a population that included girls who did not have strong achievements in or motivation toward math learning. Second, the girls in the 9th grade were typically assumed to be imbued with social stereotypic expectations, but at an age where those social forces might be counteracted in the interest of positive attitudes toward and beliefs about science and math [16]. Following the Principles and Standards of the National Council of the Teachers of Mathematics [17], the curriculum content, developed in collaboration with participating school teachers, treated fundamentals in four areas of introductory algebra, each area providing one lesson. Lesson I covered the use of real numbers - addition, subtraction, multiplication, division, and order of operations. Lesson II dealt with combining like terms and with the applications of distributive property. Lesson III covered several different methods of factoring, including using the greatest common factor, grouping, and finding the difference of two squares. Lesson IV dealt with graphing linear equations using slope and y-intercept. Each lesson, taking a one-class period and including four to five subsections, consisted of two phases: Reviews and Problem Practice. In MathGirls, the students practiced solving problems by themselves after taking lessons from their teachers in traditional settings. Figure 1 presents example screens of MathGirls. The login screen was intended to invite the girls' attention by presenting the program logo conspicuously; in the problem practice, the female teacher agent is providing corrective feedback to the student's wrong answer. The student also can read her comments below the image.

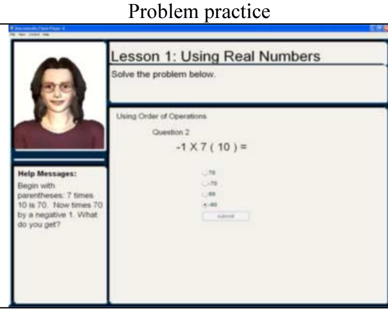
Log-in	Problem practice
	

Figure 1. Screenshots of MathGirls

1. 2 Agent Scripts

Three types of agent scripts were developed: informational, motivational, and persuasive. The informational messages were content-related, including reviews--the brief overviews of what the students had learned from their teachers--and feedback on students' performances. When a student made a mistake, the agent provided explanations to guide her to the right problem-solving path, which helped construct knowledge step by step. Motivational messages were words of praise or encouragement for the student's performance. When a student had the correct answer, the agent said "Good job" or "Great, I'm proud of you"; when the student had a wrong answer, the agent said "Everybody makes mistakes" or "You're getting there. One more thing you need to consider is…" Persuasive messages were statements about the benefits or advantages of learning math and pursuing careers in STEM. At the beginning of each section, the agent proactively made statements to positively influence the girls' perceptions of and attitudes toward math.

1.3 Pedagogical Agents Design

Four 3D virtual characters, representing male and female teachers in their forties and male and female teenagers of about 15, were designed using Poser 6 (http://www.e-frontier.com). Given the superior impact of human voices to synthesized ones, the characters' scripts were prerecorded by four voice actors matched with the age and gender of the agent. The characters and the recorded voices were integrated within Mimic Pro for lip synchronization. Facial expressions, blinking, and head movements were added to make the agents look more natural. Then the 3D animated characters were rendered in Poser to produce AVI files, which were later batch-compressed to Flash files, using Sorenson Squeeze for web casting. Figure 2 shows the four agents used in MathGirls.

Female peer	Male peer	Female teacher	Male teacher
			

Figure 2. Images of four agents

1.4 System Design

The application system was designed to accommodate four principal users: a) the students, who needed an engaging, interactive web-based experience; b) the instructional designers, who needed an easy way to create new lessons with agent videos; c) the researchers, who needed comprehensive data collection and experiment implementation; and d) the system developers, who needed a friendly programming environment for updating codes.

The developers, on the back end, connected Flash movies to dynamic instructional content with ColdFusion. The back-end relational database stored instructional texts, questions, feedback, graphics, agent videos, students' choice of agent, and students' answers to the questions. By keeping all content in the database tables, the instructional designers could update lessons and agent information using a database browser. Once the schema was set up and the basic code was written, only a few modifications were necessary to implement a new lesson. The students could start working on the lessons immediately after they logged into the system, using an Internet browser. During implementation in schools, the front-end automatically captured all responses with timestamps.

2. Research Questions

The study was guided by five specific research questions:

1) Will the presence of pedagogical agents in the learning environment influence the girls to change positively their attitude towards learning math?
2) Will the presence of pedagogical agents improve the girls' self-efficacy beliefs in learning math?
3) Will the presence of pedagogical agents influence the girls' learning outcomes?

4) Will agent characteristics (gender and age) influence the girls' choice of their agents?
5) Will agent characteristics (gender and age) influence the girls' perceptions of their agents?

3. Method

Three experimental conditions were designed: choice, randomization, and control. In the choice condition, participants were immediately given four agents, varied by gender and age (see Section 2.3), and asked to select the agent that they would like to work with. In the randomization conditions, the participants were randomly assigned to one of the four agents. In the control condition, the participants worked through the lessons without an agent, reading text-based messages.

3.1 Participants

Participants were 83 girls in required algebra classes in two high schools located in a mountain-west state of the US. Access to the samples was achieved by including four math teachers who volunteered to participate in the project. The ethnic compositions (self reported) of the samples were Caucasian (58.3%), Hispanic (22.8%), African-American (3.9%), Asian (3.3%), and others (11.7%). The average age of the participants was 15.51 (SD = 1.14).

3.2 Procedure

The project was implemented in collaboration with the math teachers in participating schools, using their regular algebra classes for two consecutive days, one lesson (approx. 50 minutes) each day. The current study implemented Lesson 1 (Using Real Numbers) and Lesson 2 (Combining Like Terms). The *MathGirls* environment was self-inclusive, students completing pretests, learning tasks, and posttests within the modules. The overall procedures were as follows:

- Teachers gave the students a brief introduction to the activity and taught them how to use the interfaces; students then put on headsets so that they could concentrate on their own tasks without distraction;
- Students accessed the web site and entered demographic information to log onto *MathGirls*, where they were randomly assigned to one of the experimental conditions (choice, randomization, and control) by system programming;
- They took pretests (self-efficacy and attitude tests only on the first day; algebra tests on both days);
- They performed the tasks (solving problems in fundamentals of algebra with guidance by the agents), in an average of 30-35 minutes; and
- They took posttests (self-efficacy and attitude tests only on the second day; algebra tests on both days. Those who worked with an agent also answered questions about agent affability on the second day).

In the lessons, the agents proactively provided information or feedback on students' performances without the students' requests. This way, students across the experimental conditions received the same amount of information.

Measures

Learning: Learning was measured by the girls' performances in pre- and posttests, with each having 10 question items. The mean scores were calculated for analysis.
Math self-efficacy: A questionnaire of six items was developed according to Bandura's guidelines [18] and was implemented before and after the intervention. The items were scaled from 1 (*Strongly disagree*) to 7 (*Strongly agree*). Item reliability evaluated with coefficient α was .83 in the pretest and .88 in the posttest. For analysis, the mean scores of the items were calculated.
Math Attitude: A questionnaire of 10 items was developed, derived from the Mathematics Attitude Survey [19] and Attitudes Toward Mathematics Inventory [20]. The girls' attitude was measured before and after the intervention. The items were scaled from 1 (*Strongly disagree*) to 7 (*Strongly agree*). Item reliability evaluated with coefficient α was .87 in the pretest and .79 in the posttest. For analysis, the mean scores of the items were calculated.
Agent Affability: A questionnaire of nine items was developed, derived from the Agent Persona Instrument [21]. The items were scaled from 1 (*Strongly disagree*) to 7 (*Strongly agree*). Item reliability evaluated with coefficient α was .90. For analysis, the mean scores of the items were calculated.
Choice of an agent: Learners' choice was recorded by the program.

To analyze learning, math self-efficacy, and math attitude, Paired *t*-tests with group membership were conducted focused on each research question. To analyze agent affability, one-way ANOVA (by agent age and gender) was conducted. To analyze learners' choice of an agent, Chi-square analysis (χ^2) was conducted. For all the analyses, the significant level was set at $\alpha <.05$.

4. Results

4.1 Learning

The results revealed that all the participants, regardless of agent presence, significantly increased their math performances after working at the MathGirls environment: for the girls working with the agents, $t = 4.32$, $p < .001$, $d = .41$; and for the girls without an agent, $t = 2.70$, $p < .01$, $d = .50$. The effect sizes (Cohen's *d*) indicated medium effects by Cohen's guidelines.

4.2 Math Self-Efficacy

The results revealed that only for the girls who worked with agent was there a significant increase from the pre-test ($M = 26.75$, $SD = 7.21$) to the post-test math self-efficacy ($M = 29.25$, $SD = 7.27$), $t = 2.44$, $p < .01$, $d = .35$. The effect size of this difference indicated a small effect by Cohen's guidelines. Further, the detailed analyses revealed that the girls who worked with peer agents significantly increased their self-efficacy from the pre-test ($M = 27.3$, $SD = 5.3$) to the post-test ($M = 30$, $SD = 6.64$), $t = 2.16$, $p < .05$, $d = .44$. The effect size (*d*) was medium. For the girls who worked without an agent, there was no significant difference between the pre- and posttest math self-efficacy.

4.3 Math Attitude

The results revealed that only for the girls who worked with an agent was there a significant increase from the pre-test (M = 44.16, SD = 12.21) to the post-test math attitude (M = 46.93, SD = 9.88), t = 1.96, p < .05, d = .25. For the girls who worked without an agent, there was no significant difference between the pre- and post-tests math attitude.

4.4 Agent Affability

One way ANOVA revealed a significant difference between the girls who worked with peer agents (M = 45.57, SD = 8.92) and those who worked with teacher agents (M = 40.05, SD = 10.94), F = 4.95, p < .05. The girls perceived the peer agents to be significantly more affable than the teacher agents.

4.5 Choice of an Agent

The results revealed a significant difference in the girls' choice of their agents, χ^2 = 10.36, p < .05. Forty-eight percent of the girls chose the female peer agent; 32% chose the female teacher agent; 12% chose the male peer agent; and 8% chose the male teacher agent. That is, a total of 80% of the girls chose female agents as their learning partners.

5. Discussion

The study investigated the potential of pedagogical agents to positively influence high school girls' math attitude, math self-efficacy beliefs, and math learning and the potential of the agents to play differing social roles, depending on their characteristics (i.e., gender and age). First, the results overall supported the instructional value of agent presence in the learning environment. The girls who listened to the agent's messages significantly developed their positive attitude towards and increased their self-efficacy beliefs in learning math, whereas the girls who read text-based messages in the control condition did not. Also, the girls chose female agents significantly more than male agents and perceived peer agents as significantly more affable than teacher agents. These findings, in line with Bandura's concept of attribute similarities in the traditional classrooms, confirm learners' tendency to perceive virtual characters socially. The uniqueness of the study, however, might be rigorously proving the value of agents in adding social richness to the environment in terms of this specific context: math-related attitude and self-efficacy beliefs of high school girls. Further, after working with MathGirls, the girls significantly increased their learning, regardless of agent presence, again confirming the current status of knowledge that agent presence has worked effectively for learners' affect but not for their learning outcomes. Three final cautions: First, the variations in the four voice actors were not controlled, which might affect the results. Second, the duration of the study was only two days. We may want to determine whether the immediate results endure over time. Third, learner attitude and self-efficacy were measured only by self-report. Other measures might be considered. These cautions may help guide future directions of this research. This study could be extended to examine long-term effects, self-reports might be complemented with behavioral indicators, and the girls' reactions to the agents could be compared with the reactions of a matched group of boys. In short, much still remains to be learned.

Acknowledgements

This work was sponsored by the National Science Foundation (GSE-05226343).

References

[1] Y. Kim and A. L. Baylor, "A social-cognitive framework for designing pedagogical agents as learning companions," *Educational Technology Research & Development*, vol. 54, pp. 569-596, 2006.

[2] I. Arroyo, T. Murray, B. P. Woolf, and C. R. Beal, "Further results on gender and cognitive differences in help effectiveness," presented at International Conference of Artificial Intelligence in Education, Sydney, Australia, 2003.

[3] E. M. Evans, H. Schweingruber, and H. W. Stevenson, "Gender differences in interest and knowledge acquisition: The United States, Taiwan, and Japan," *Sex Roles*, vol. 47, pp. 153-167, 2002.

[4] K. Hakkarainen and T. Palonen, "Patterns of female and male students' participation in peer interaction in computer-supported learning," *Computers & Education*, vol. 40, pp. 327-342, 2003.

[5] V. Slotte, K. Lonka, and S. Lindblom-Ylanne, "Study-strategy use in learning from text. Does gender make any difference?" *Instructional Science*, vol. 29, pp. 255-272, 2001.

[6] A. L. Zeldin and F. Pajares, "Against the odds: Self-efficacy beliefs of women in mathematical, scientific, and technological careers," *American Educational Research Journal*, vol. 37, pp. 215-246, 2000.

[7] I. Arroyo, J. E. Beck, B. P. Woolf, C. R. Beal, and K. Schultz, "Macroadapting animalwatch to gender and cognitive differences with respect to hint interactivity and symbolism," in *Intelligent Tutoring Systems, Proceedings*, vol. 1839, *Lecture Notes in Computer Science*, 2000, pp. 574-583.

[8] D. Passig and H. Levin, "Gender preferences for multimedia interfaces," *Journal of Computer Assisted Learning*, vol. 16, pp. 64-71, 2000.

[9] A. L. Baylor and Y. Kim, "The role of gender and ethnicity in pedagogical agent perception.," presented at E-Learn, the Annual Conference of Association for the Advancement of Computing in Education., Phoenix, AZ, 2003.

[10] P. Persson, J. Laaksolahti, and P. Lonnqvist, "Understanding social intelligence," in *Socially intelligent agents: Creating relationships with computers and robots*, K. Dautenhahn, A. H. Bond, L. Canamero, and B. Edmonds, Eds. Norwell, MA: Kluwer Academic Publishers, 2002.

[11] Y. Kim, "Pedagogical agents as learning companions: Building social relations with learners," presented at 12th International Conference on Artificial Intelligence in Education, Amsterdam, The Netherlands, 2005.

[12] A. Bandura, *Self-efficacy: The exercise of control*. New York: W. H. Freeman., 1997.

[13] D. H. Schunk, "Peer models and children's behavioral change," *Review of Educational Research*, vol. 57, pp. 149-174, 1987.

[14] Y. Kim, "Learners' expectations on the desirable characteristics of learning companions," presented at the Annual Conference of American Educational Research Association, San Diego, CA, 2004.

[15] Y. Kim, "Pedagogical agents as learning companions: The effects of agent affect and gender on learning, interest, self-efficacy, and agent persona," in *Educational Psychology and Learning Systems*, vol. PhD. Tallahassee, FL: Florida State University, 2004.

[16] P. H. Dunham, "Procedures to increase the entry of women in mathematics-related careers," in *ERIC/SMEAC Mathematics Education Digest No. 3*, 1990.

[17] NCTM, "Principles and standards for school mathematics," 2000.

[18] A. Bandura, "Guide for constructing self-efficacy scales," vol. 2004, 2001.

[19] C. A. Ethington and L. M. Wolfe, "Women's selection of quantitative undergraduate fields of study: Direct and indirect influences," *American Educational Research Journal*, vol. 25, pp. 157-175, 1988.

[20] M. Tapia and G. E. Marsh, "An instrument to measure mathematics attitudes," vol. 2004, 2004.

[21] A. L. Baylor and J. Ryu, "The API (Agent Persona Instrument) for assessing pedagogical agent persona," presented at International Conference of Ed-Media, Honolulu, Hawaii, 2003.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Content Matters: An Investigation of Feedback Categories within an ITS

G. Tanner JACKSON[1], Arthur C. GRAESSER
Department of Psychology, University of Memphis, Memphis, TN, USA

Abstract. The primary goal of this study was to investigate the role of feedback in an intelligent tutoring system (ITS) with natural language dialogue. One core component of tutorial dialogue is feedback, which carries the primary burden of informing students of their performance. AutoTutor is an ITS with tutorial dialogue that was developed at the University of Memphis. This article addresses the effectiveness of two types of feedback (content & progress) while college students interact with AutoTutor on conceptual physics. Content feedback provides qualitative information about the domain content and its accuracy as it is covered in a tutoring session. Progress feedback is a quantitative assessment of the student's advancement through the material being covered (i.e., how far the student has come and how much farther they have to go). A factorial design was used that manipulated the presence or absence of both feedback categories (content & progress). Each student interacted with one of four different versions of AutoTutor that varied the type of feedback. Data analyses showed significant effects of feedback on learning and motivational measures, supporting the notion that "content matters" and the adage "no pain, no gain."

Keywords: Pedagogical Feedback, Intelligent Tutoring Systems

Introduction

It has long been known that one-to-one human tutoring is a highly effective form of instruction when compared with traditional classroom settings [1,2]. One of the interesting findings is that even inexperienced tutors, untrained in pedagogy [3], still significantly increase student learning [2,4]. This finding raises a fundamental question: Which components of tutoring are responsible for the significant learning gains? Researchers have tested many tutoring environments, both human and computer, in an attempt to answer this question.

Tutorial feedback is presumably a vital component in the process of tutorial dialogue, learning outcomes, and the metacognitive states of the learner. Many aspects of tutorial feedback have been studied within the realm Intelligent Tutoring Systems (ITS): feedback content, feedback latency, control of feedback, purpose of feedback, and so on [5]. ITS researchers have recently incorporated tutorial dialogue in natural language in their investigations of feedback, with feedback being implemented at varying grain sizes [6,7,8]. Researchers at the Institute for Intelligent Systems at the University of Memphis have created an ITS called AutoTutor that uses natural language to tutor students in various domains (computer literacy, conceptual physics,

[1] Corresponding Author: G. Tanner Jackson, Department of Psychology, University of Memphis, Memphis, TN, 38152, USA; E-mail: gtjacksn@memphis.edu.

and critical thinking). Recent work on AutoTutor has examined potential dialogue components, including feedback, that contribute to overall learning gains [9]. The present study investigated the effectiveness of two types of feedback while college students interact with AutoTutor on conceptual physics.

1. Two Types of Feedback: Content and Progress

This study focuses on two particular categories of feedback: content feedback and progress feedback. Content feedback provides qualitative information about the domain content and its accuracy as it is covered in a tutoring session. Progress feedback is a quantitative assessment of the student's advancement through the material being covered (i.e., how far the student has come and how much farther they have to go). These two categories of feedback are not mutually exclusive and sometimes coexist to varying degrees. However, they will be considered separately for the current purposes.

Almost all of the work on feedback within the ITS community has focused on the important parameters surrounding *content* feedback. This includes assessing the accuracy of the students' information, flagging and/or remediating any errors when they are found, the timing of feedback presentation, and other domain-specific computations [5,10]. Research by Anderson and colleagues (1995) manipulated the control of feedback (controlled by tutor versus sought by student) and the information included in the feedback (e.g., identifying an error versus identifying and including a possible explanation/solution). They found that including an explanation or solution to an error increased knowledge retention after a delay.

The previous research on content feedback has typically demonstrated its importance. This body of previous research has recently been synthesized in an article by Schute (2006), in which she attempts to organize the scientific findings into some modicum of a coherent framework [11].

Contrasted with content feedback, *progress* feedback simply provides the student with an approximate measure of how well they are moving forward through the learning process. Tutors often unknowingly produce this kind of feedback when they say something like, "alright, you've completed several practice problems so let's do two more and then we can move on". This simple statement lets the student know that they have been working on this concept for a while and that they will only have a few more things to try before they get a break. This progress feedback provides the student a sense of accomplishment and gives them an idea of how much they have left to do [12]. Most research on progress feedback has fallen within the education and survey methodology literature.

Progress feedback has been shown to increase motivation and the desire to improve performance through an increase in self-efficacy. The concept of self-efficacy refers to a person's beliefs about their own capacity to learn or perform at specific levels. Previous research in school settings has shown that there is an increase in student self-efficacy as a function of increases in attention to individual student progress [13]. On the flip side, as students have fewer feelings of making progress, they begin to drop in levels of self-efficacy. Schunk and colleagues have demonstrated that there is an associated increase in student self-efficacy and motivation to continue improving performance by providing feedback on the overall progress through learning [12]. These empirical findings support the possibility that educational games that

combine motivation, fine-grained feedback, and serious content can have a major impact on the educational enterprise.

Recent research by Conrad, Couper, Tourangeau, and Peytchev (2005) has investigated the use of progress feedback within survey methodology [14]. These researchers have reported that respondents often make an initial and lasting impression of their progress while answering a set of questions. A second study reported in Conrad et al. (2005) suggested that an intermittent display of progress in the late phase of an experiment may provide the most benefits and fewest costs, when compared to steady progress or on-demand progress feedback.

Progress feedback has received very little or no attention in the ITS field, yet it may play a vital role in developing longer-term learning relationships between computer tutors and learners. This feedback may be needed to help encourage return visits with learning systems and to foster user's self-efficacy that will ultimately lead to improved learning.

2. AutoTutor

AutoTutor has been described in other publications [6,8,15], so it will only be briefly discussed here. AutoTutor is an ITS with an intelligent agent that can deliver conversational dialogue that ideally is both pedagogically effective and engaging. The researchers and designers of AutoTutor have developed a tutorial model that simulates novice tutoring strategies [16]. The tutorial model includes theories of pedagogical strategies as well as naturalistic dialogue. Unlike humans, AutoTutor can manipulate the tutorial dialogue in a controlled and systematic manner at both fine-grained and global levels. AutoTutor makes use of an animated conversational agent with facial expressions, synthesized speech, and rudimentary gestures. AutoTutor is more than an information delivery system. It is a collaborative scaffold that uses natural language conversation to assist students in actively constructing knowledge.

AutoTutor utilizes strategies that are common to human tutors, such as expectation tailored dialogue. Most human tutors work with students through a problem and look for particular correct answers (called *expectations*) and particular misunderstandings (*misconceptions*). There are typically multiple expectations and misconceptions associated with each problem. As a learner articulates an answer or solves a problem, the tutor is constantly comparing the student's answer content with the expectations and misconceptions. AutoTutor performs this same comparison and uses it to adaptively and appropriately respond whenever particular expectations or misconceptions arise. This tutoring process is referred to as *expectation and misconception tailored dialogue* (EMT dialogue) [17].

AutoTutor utilizes Latent Semantic Analysis (LSA) and other conceptual pattern matching algorithms to make the EMT dialogue comparisons for each learner contribution. LSA is a high-dimensional, statistical analysis that uses vector quantities to represent general semantic knowledge. The similarity of content is represented as a cosine, a numerical value that typically ranges from 0 (not at all similar) to 1 (exactly the same). These vectors are used to calculate the conceptual similarity of any two segments of text, which could be as small as a word or as large as a complete document [18]. For example, a target piece of text (i.e., a student contribution) is selected and LSA compares it to another bit of text (i.e., one of the expectations from the current problem). LSA calculates the conceptual similarity between student contributions and

expected good answers (and also to possible misconceptions). AutoTutor uses the LSA value to determine if expectations (or misconceptions) can be considered covered and to determine which is the next appropriate move based on a given pedagogical strategy.

AutoTutor has been evaluated on learning gains in a collection of studies. These evaluations have continually shown significant learning gains for the students who interact with the tutor, with effect sizes of approximately .8 standard deviation units [6,8]. Given that we have empirically established AutoTutor's effectiveness on learning gains, we have extended our manipulations of interest to examine potential causes of the learning.

3. A Study that Compares Content and Progress Feedback

As discussed earlier, the ITS research on feedback has largely focused on aspects of content feedback, but little or no attention to progress feedback. Available ITS research has provided no direct comparisons between the cognitive components of content feedback with the motivational components of progress feedback. Such a comparison would be desired to better understand how tutoring can provide an effective, efficient, and possibly long-term learning experience. In some sense, the present study is comparing cognition and motivation. Our focus on the two types of feedback (content versus progress) decomposes each kind of feedback into both local and global levels (which will not be directly compared in this paper). Local levels allow more fine-grained and immediate feedback; these occur frequently. In contrast, the global levels provide a more coherent or overall sense of feedback; these occur occasionally. Each level of feedback should provide a unique contribution to the learning experience.

3.1. Procedure

The experiment had a factorial design that manipulated the presence or absence of content feedback and the presence or absence of progress feedback. The experiment consisted of three phases: pretest phase, training phase, and posttest phase.

During the pretest phase, participants completed a demographics survey followed by a 25 item pretest. The pretest consisted of 25 multiple choice physics questions, either pulled from or similar to the Force Concept Inventory (FCI). The FCI is an established test in the area of conceptual physics [19].

The training phase involved the participants working with AutoTutor through four conceptual physics problems in one of four different conditions. These conditions are discussed in more detail below. All versions of AutoTutor covered the same four problems and utilized the same working algorithms as previous versions of AutoTutor.

The posttest phase included two sets of questions. The posttest assessed conceptual physics knowledge through a different set of 25 questions (counterbalanced across participants with the set of 25 physics questions from the pretest). The second set of questions addressed levels of motivation and users' attitudes towards the tutor (13 questions).

3.2. Experimental Conditions.

A 2x2 factorial design was used to create four conditions with different combinations of feedback components. In this study, local and global levels of feedback were combined (rather than separated) so all feedback variations treated both local and global levels together. The four conditions consisted of content and progress feedback together (screenshot in Figure 1), content feedback only, progress feedback only, and no feedback.

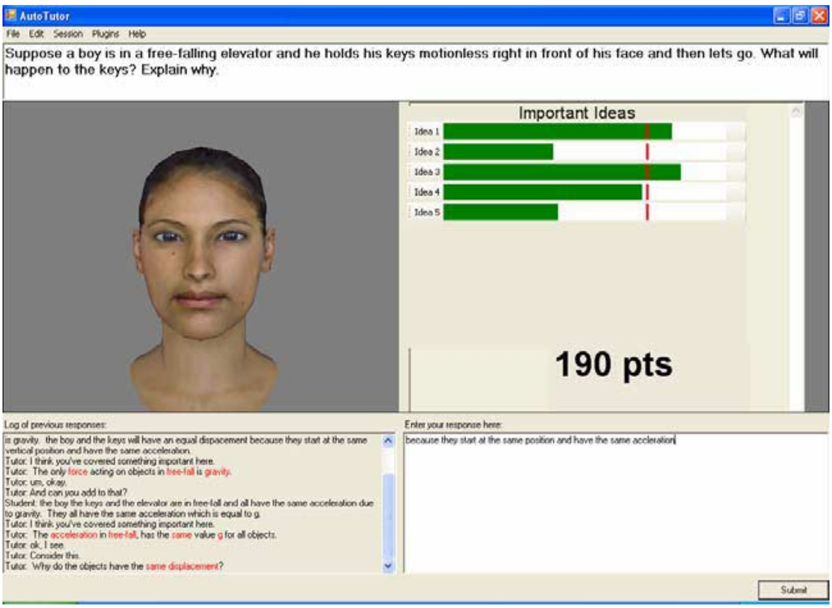


Figure 1. AutoTutor in the Content+Progress Feedback condition.

Content+Progress Condition. This condition had both content and progress feedback. Figure 1 shows a screen shot of this condition. Regarding the content feedback, the global level of content feedback consisted of a summary of the solution to a problem, after the AutoTutor-student dialogue on the problem was completed. The local content feedback consisted of the coloring of important words within a dialogue history box that was presented on the computer screen. All dialogue (either spoken by the tutor or typed in by the student) was presented on the screen in a dialogue history box (lower-left corner of Figure 1). In each expectation, there were several important words required to understand the concept (e.g., horizontal velocity, free-fall, gravity). In the dialog history window these important words were presented in red (rather than black) to distinguish the important concepts stated by the tutor.

The Content+Progress condition also included both levels of progress feedback. The global progress feedback consisted of a game points system, where students were awarded points depending on their level of progress through each problem (i.e., how quickly they covered certain expectations within a problem). The local level of progress feedback included the addition of progress bars, which visually represented the level of progress for each individual expectation within a problem (one bar for each expectation). The value displayed in each bar directly corresponded to the latent semantic analysis (LSA) value for each respective expectation in that problem.

Content-only Condition. The content feedback condition included both global and local levels of content feedback only. This condition included summaries and colored words, but did not include either type of progress feedback.

Progress-only Condition. This condition included local and global levels of progress feedback, but excluded both types of content feedback. That is, game points and progress bars were displayed, but answer summaries and highlighted words were excluded.

No Feedback Condition. This condition included none of the aforementioned types of feedback.

4. Results

A total of 83 students were included in the following analyses. Students' raw performance on the pretest and posttest were converted into proportion scores (see Table 1 for means). These proportion scores were used to calculate two learning measures. Simple learning gains were computed for each student by subtracting their pretest proportion from their posttest proportion. This provided a direct measure of improvement from pre- to post-test. One problem with the simple learning gains is that they are biased towards students with low pretest scores because they have more room for improvement. Therefore, the proportional learning gains were computed as an additional learning measure: [(post-test – pretest)/(1-pretest)]. Means for simple and proportional learning gains for each condition are presented in Table 1.

Table 1. Means (SD) on learning measures for all 4 conditions (n=83)

Conditions	**Pretest Proportion**	**Posttest Proportion**	**Simple Learning Gains**	**Proportional Learning Gains**
Nothing	.38 (.11)	.43 (.16)	.05 (.13)	.07 (.23)
Progress Feedback	.41 (.11)	.48 (.17)	.08 (.12)	.14 (.21)
Content Feedback	.33 (.09)	.51 (.17)	.19 (.16)	.28 (.25)
Content & Progress Feedback	.35 (.12)	.48 (.13)	.13 (.12)	.19 (.17)

4.1. Learning Measures

An ANOVA with a 2x2 factorial design was performed on each of the learning measures. There was a statistically significant main effect of content feedback for both simple learning gains, $F(1,79) = 10.83$, $p < .05$, $MS_e = .018$, and for proportional learning gains, $F(1,79) = 7.20$, $p < .05$, $MS_e = .047$. However, the main effect of progress feedback was not statistically significant for simple learning gains, $F(1,79) = 0.22$, $p > .50$, $MS_e = .018$, and for proportional learning gains, $F(1,79) = 0.03$, $p > .50$, $MS_e = .047$. There were no significant interactions between content and progress feedback, $F(1, 79)= 2.30$, $p > .10$, $MS_e = .018$, and $F(1, 79)= 2.43$, $p > .10$, $MS_e = .047$, respectively. These results are consistent with the conclusion that it is the content feedback that matters, not the progress feedback.

4.2. Interest and Motivation Measures

Analyses performed on the motivation and interest measures revealed an interesting contrast to those from the learning measures. All motivational and interest measures were rated on a six-point scale with higher numbers reflecting an increase in agreement (e.g., a higher score for motivation means they felt more motivated). An ANOVA with a 2x2 factorial design was performed on each of the 13 motivational and interest measures. There was a statistically significant main effect of content feedback (lower ratings for those exposed to content feedback) for both interest in the tutoring experience, $F(1, 79)= 7.05$, $p < .05$, $MS_e = 2.284$, and for feeling motivated, $F(1, 79)= 6.651$, $p < .05$, $MS_e = 2.355$. The significant findings presented here are representative of an overall trend in which students who received some form of content feedback (either content only or the combination of content and progress feedback) actually rated most motivation and attitude measures lower than those students who did not receive any form of content feedback (progress only or no feedback). The main effect of progress feedback was not significant for any of the measures, including interest in the experience, $F(1, 79)= 0.22$, $p > .60$, $MS_e = 2.284$, and motivation $F(1, 79)= 0.00$, $p > .90$, $MS_e = 2.355$. The interaction between content and progress feedback was not significant for either interest in the tutoring experience, $F(1, 79)= 3.22$, $p > .05$, $MS_e = 2.284$, or feeling motivated, $F(1, 79)= 0.86$, $p > .30$, $MS_e = 2.284$. The significant main effect revealed that students who interacted with some form of content feedback actually rated their interest and motivation levels as lower than students who interacted with either progress feedback or no feedback at all.

5. Discussion and Implications

The results from the learning outcomes clearly support the inclusion of content feedback. Students who received at least some form of content feedback benefited more from the tutoring than those who received either progress or no feedback. Simply put, it is the content that matters, a purely cognitive variable, when considering the impact of feedback on learning. Progress feedback did not significantly affect learning.

The results from the motivational and attitudinal measures will be counterintuitive to some researchers and instructors. The exposure to content feedback yielded lower ratings, whereas it is not the case that exposure to progress feedback increased motivation ratings (the no feedback condition actually had the highest scores). The motivation and attitude of the learners was inversely related to the occurrence of feedback, and particularly the content feedback where they learned the most.

Taking into account the results from both learning and motivational measures we come away with the provocative picture that can be captured in the adage, “no pain, no gain”. These data clearly demonstrate an inverse relationship between liking a tutoring experience and how much that gets learned. The learning outcomes demonstrate that the content feedback provides the most effective learning experience, whereas the motivational measures tend to show a dislike for content feedback. And of course, a dislike of the learning experience would be expected to decrease the likelihood of a return visit to the tutor for those students who do not want to be challenged. As researchers and developers, we are left in a quandary: We want to provide the most productive learning experiences possible, yet we also want to build an environment that learners enjoy using and would be willing to use again. This represents an ongoing

struggle between liking and learning for ITS researchers. Somehow we need to settle on a delicate balance.

Acknowledgements

This research was supported by the National Science Foundation (ITR 0325428) and the Institute of Education Sciences (R305H050169). Any opinions, findings and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of NSF and IES.

References

[1] Bloom, B.S. (1984). The 2 sigma problem: the search for methods of group instruction as effective as one-to-one tutoring. *Educational Researcher*, 13, 4-16.
[2] Cohen, P.A., Kulik, J.A., & Kulik, C.L.C. (1982). Educational outcomes of tutoring: a meta-analysis of findings. *American Educational Research Journal*, 19, 237-248.
[3] Fitz-Gibbon, C.R. (1977). *An analysis of the literature of cross-age tutoring*. Washington, DC: National Institue of Education. (ERIC Document Reproduction Service No. ED 148 807).
[4] Fantuzzo, J.W., King, J.A., & Heller, L.A. (1992). Effects of reciprocal peer tutoring on mathematics and school adjustment: a component analysis. *Journal of Educational Psychology*, 84, 331-339.
[5] Anderson, J.R., Corbett, A.T., Koedinger, K.R., & Pelletier, R. (1995). Cognitive tutors: lessons learned. *Journal of the Learning Sciences*, 4, 167-207.
[6] Graesser, A.C., Lu, S., Jackson, G.T., Mitchell, H., Ventura, M., Olney, A., & Louwerse, M.M. (2004). AutoTutor: A tutor with dialogue in natural language. *Behavioral Research Methods, Instruments, and Computers, 36*, 180-193.
[7] Jordan, P & VanLehn, K. (2002) Discourse processing for explanatory essays in tutorial applications. Proceedings of 3rd SIGdial Workshop on Discourse and Dialogue, (pp. 74-83). Philadelphia, PA.
[8] VanLehn, K., Graesser, A.C., Jackson, G.T., Jordan, P., Olney, A., & Rose, C.P. (in press). When are tutorial dialogues more effective than reading? *Cognitive Science*.
[9] Jackson, G.T., & Graesser, A.C. (2006). Applications of human tutorial dialog in AutoTutor: An intelligent tutoring system. Revista Signos, 39, 31-48.
[10] Koedinger, K. R., Anderson, J. R., Hadley, W. H., & Mark, M. A. (1997). Intelligent tutoring goes to school in the big city. *International Journal of Artificial Intelligence in Education,* 8, 30-43.
[11] Shute, V. J. (2006). *Focus on formative feedback.* ETS Research Report, Princeton, NJ.
[12] Schunk, D. H., & Pajares, F. (2001). The Development of Academic Self-Efficacy. In A. Wigfield & J. Eccles (Eds.), *Development of Achievement Motivation.* San Diego: American Press.
[13] Pintrich, P. & Schunk, D. (1996). *Motivation in Education: Theory, Research & Applications*, Ch. 3. Englewood Cliffs, NJ: Prentice-Hall.
[14] Conrad, F., Couper, M., Tourangeau, R., Peytchev, A. (2005). Impact of progress feedback on task completion: first impressions matter. *Proceedings of SIGCHI 2005: Human Factors in Computing Systems.* Portland, OR. 1921-1924.
[15] Graesser, A.C., Wiemer-Hastings, K., Wiemer-Hastings, P., Kreuz, R., & the TRG (1999). Auto Tutor: A simulation of a human tutor. *Journal of Cognitive Systems Research, 1*, 35-51.
[16] Graesser, A.C., Person, N.K., & Magliano, J. (1995). Collaborative dialog patterns in naturalistic one-on-one tutoring. *Applied Cognitive Psychology*, 9, 495-522.
[17] Graesser, A.C., Hu, X., & McNamara,D.S. (2005). Computerized learning environments that incoporate research in discourse psychology, cognitive science, and computational linguistics. In A.F. Healy (Ed.), *Experimental Cognitive Psychology and its Applications: Festschrift in Honor of Lyle Bourne, Walter Kintsch, and Thomas Landauer.* Washington, D.C.: American Psychological Association.
[18] Landauer, T., Foltz, P. W., & Laham, D. (1998). An introduction to latent semantic analysis. *Discourse Processes*, *25*, 259-284.
[19] Hestenes, D., Wells, M., & Swackhamer, G. (1992). Force Concept Inventory. *The Physics Teacher. 30.* 141-158.

Does Learner Control Affect Learning?

Joseph E. BECK
Machine Learning Department, Carnegie Mellon University
joseph.beck@gmail.com

Abstract. Many intelligent tutoring systems permit some degree of learner control. A natural question is whether the increased student engagement and motivation such control provides results in additional student learning. This paper uses a novel approach, learning decomposition, to investigate whether students do in fact learn more from a story they select to read than from a story the tutor selects for them. By analyzing 346 students reading approximately 6.9 million words, we have found that students learn approximately 25% more in stories they choose to read, even though from a purely pedagogical standpoint such stories may not be as appropriate as those chosen by the computer. Furthermore, we found that (for our instantiation of learner control) younger students may derive less benefit from learner control than older students, and girls derive less benefit than boys.

1 Introduction

Many intelligent tutoring systems (ITS) have some element of learner control. For example, the Reading Tutor [1] sometimes allows students to select which story they will read next. Furthermore, most tutoring systems allow students to determine when to ask for help (e.g. [2] and see [3]). Learner control is certainly helpful in making students feel more engaged with their learning, and engaged learners tend to learn more [4]. Presumably, the transitive argument holds that allowing learner control will result in more student learning. However, there are possible drawbacks to learner control. First, by their very nature students are not expert in the content to be learned. While it might make sense to allow a thirty-five year old who is using a computer based training system to study for an exam for promotion to control his own learning, it is far less obvious that allowing a six year old such control will be as productive. Students do not always make optimal choices for their learning [3]. The notion of "perceived learner control" [5] avoids these problems. In this framework, the choices students are given do not have large implications for the actual domain content seen. For example, students could choose which icon will be used to represent progress (e.g. [6]). The second drawback of learner control is that computers can make decisions much more quickly than can students. Time spent letting the student make decisions is time spent not learning the material. Given the primacy of time on task for learning, this trade should be made with extreme caution.

This paper addresses the following questions: Does perceived learner control influence how much students learn? By what amount? Which students benefit? Are any gains simply from novelty or are they persistent? And does the benefit from allowing students to influence their learning outweigh the time lost from learning activities that such interactions entail?

2 Approach

In Project LISTEN's Reading Tutor [1], the student and tutor take turns picking which story the student is going to read next. As shown in Figure 1, when it is the student's turn to select a story, the tutor presents four stories and how many times the student has read that story previously. The student then clicks on whichever story he wishes to read. If the student does not like the four stories presented, he can select to see other stories at the same level of difficulty, or he can select stories at a different level of difficulty. To avoid the stigma of being behind one's peers, the story levels are somewhat awkwardly encoded: "K" represents the lowest story level (short for "kindergarten"), followed by "A," then "B," up to level "G." Although the interface may seem somewhat baroque, finding a design that works for young children who are barely able to read is challenging. See [7] for more details of the interface for selecting stories. When it is the Reading Tutor's turn to pick a story, the tutor randomly selects a story at the student's current reading level that he has not yet read.

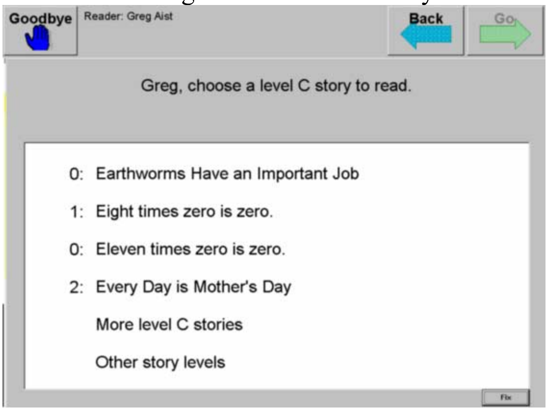


Figure 1. Student story choice menu in the Reading Tutor.

Once the student selects a story to read, the tutor presents it on the screen one sentence (or, for very long sentences, sentence fragment) at a time. Students read aloud to the tutor and are able to ask for help for challenging words. If the student does not like the story he is reading, he can return to the story choice menu to get a different story.

One method of determining whether the Reading Tutor's learner control is beneficial is to run a controlled study where one group of students always selects which story to read and the other group always has their story selected by the tutor. There are a few problems with this methodology. First, our *a priori* hypothesis is that there will be a relatively small effect on learning from students being allowed to select their own story to read. Therefore, a study would have to involve many users, be of considerable duration, or some combination of both, and would consequently be expensive to run.

Second, who would be the consumer for the results of such a study? The effect of learner control depends both on the types of learners and on the type of control. Learners with weaker metacognitive skills will (presumably) show less benefit from control than metacognitively advanced learners. Furthermore, perhaps some types of control are more effective than others. Would the results of such a study be informative for high school students studying geometry or for college students learning physics? Even if the learners were similar, would the results transfer across different types of learner control and to different domains?

Given the limited utility to other researchers, the option of running an expensive study is not an appealing one. Rather than trying paying a lot to get a definitive answer to a question that few others care about (how effective is the *Reading Tutor's* learner control?), instead we will concentrate on creating an easy recipe to enable researchers to answer the question of how effective is *their tutor's* learner control. If any researchers who were curious could cheaply test whether their system's learner control was beneficial, then enough such results might give us a better understanding of what types of control—and for what populations of students—there is a benefit. Even if the specific results for the effects of learner control in the Reading Tutor were not of interest to others, our method of obtaining the answer would be.

2.1 Learning Decomposition

To measure the impact of learner control, we use *learning decomposition* [8]. The idea of learning decomposition is to modify the learning curve equation to account for the impact of different types of learning opportunities. As can be seen in the standard form of the exponential learning curve in Equation 1a, the free parameter A represents how well students perform on their first trial performing the skill; e is the numerical constant (2.718); the free parameter b represents how quickly students learn the skill, and t is the number of practice trials the learner has had at this skill. This model can be solved with any non-linear regression package (we use SPSS 11.0). For this work, we are tracking student knowledge at the level of individual words, and therefore trials (t) are every time the student encounters the word in a story.

Traditionally, learning curves simply count up the number of prior learning opportunities. Learning decomposition instead separates learning opportunities into different types, as specified by the researcher. Equation 1b shows the form of the learning decomposition model. A, e, and b have the same meanings as in Equation 1a. Rather than simply counting all prior opportunities (t) equally, we split prior exposures into those that occurred in stories chosen by the Reading Tutor (t_1) and those chosen by the student (t_2). Equation 1b shows this decomposition. Since either the student or the tutor chose every story, it is the case that $t = t_1 + t_2$.

The additional free parameter, B, represents the relative impact of tutor control. If, for example, the best fitting parameter estimate for B is 0.5, then the development of student proficiency is best described by a model that weights learning opportunities in tutor selected stories as being worth half as much as opportunities in tutor selected stories. I.e. students learn half as quickly when the tutor selects the story they read. Note that our choice to allow the value of tutor selected stories to vary has no impact on the analysis. If we had instead estimated the effect of student selected stories by modeling $t_1 + B*t_2$ then we would have found that B=2.0 (i.e. students learn twice as quickly when they select the story—an identical finding). For our analysis, if $B>1$ then tutor choice results in additional learning over student control. If $B=1$, then there is no difference in learning. If $B<1$ then students learn more in stories that they pick.

Equation 1b are abridged for clarity, as we also consider the length of the word and how many prior help requests the student has made on the word. We will not discuss those aspects of the model in this paper.

$$performance = A * e^{-b*t} \qquad performance = A * e^{-b*(B*t_1+t_2)}$$

Equation 1. (a) Exponential model, **(b)** Learning decomposition model of practice.

2.2 Measure of Reading Growth

We now instantiate what we mean by *performance* in Equation 1. When students read a word in the Reading Tutor, we record how long they took to read the word, whether the automated speech recognizer (ASR) thought the word was read correctly, and whether they asked for help on the word. We would like a performance measure to improve smoothly over time. Therefore, reading time is a better measure than accuracy or help request, both of which are binary outcomes and thus incapable of gradual improvement on individual trials. However, reading time is undefined for cases where the student did not attempt to read the word, and is artificially lowered for cases where the student first asks for help since the student can simply mimic the tutor's help. To avoid these confounds, we set reading time to be 3.0 seconds for cases where the student skips the word or asks for help. The time of 3.0 seconds is fairly high, since only 0.1% of readings exceeded that amount. Furthermore, reading times that exceeded 3.0 seconds were reduced to be 3.0 seconds. Although speech recognition is far from perfect and individual estimates of word reading time are inaccurate, errors should be randomly distributed. I.e. we do not expect either type of trial to benefit unfairly from the scoring inaccuracies and the results are an unbiased estimate of learner control.

We would also like our performance metric to reflect changes in student knowledge rather than temporary scaffolding effects. For example, if the student reads a story that has the word "divided" in 5 consecutive sentences (such stories exist in the Reading Tutor), by the fifth sentence one suspects the student is no longer reading the word in the normal sense, but is simply retrieving it from short term memory. Therefore, we only count the student's first exposure to a word in a day for the dependent variable. However, we count all of the exposures as possible learning opportunities (i.e. for the independent variable). So, in the above example, only the first sentence with "divided" would count as a training opportunity for the model, but all of the sentences would be counted to determine the correct values for t_1 and t_2. Fitting a simple exponential model that uses number of prior exposures to predict mean word reading time for a word with that many exposures had an R^2 of 76%. So our data are well described by an exponential model.

Our approach was to fit the nonlinear regression model in Equation 1b to each individual student. That is, we estimated the A, b, and B parameters for every student using only his data, and only considering students who read at least 20 words in the Reading Tutor. We used every word read by the student as training data for the model. We accounted for varying length of words by including a simple linear term to account for word length in characters. We expect considerable error in the parameter estimates of B, and so cannot make claims about the efficacy of learner control for individual students, but can use the noisy estimates to detect overall patterns across students. Our resulting data set had 346 students who had a total of 959,455 observations of learning (that is, first attempts for a student at reading a word that day). Overall these students read approximately 6.9 million words over the course of the year.

3 Results

To determine the effects of tutor- vs. student-selected stories, we compute the median B parameter estimates for all 346 students. The median value of B was 0.80, i.e. students only learned 80% as much in stories chosen by the tutor (alternately, 25% more in

stories selected by the student). Note that this measure of increased learning does not consider short-term motivational effects. First, if a student read a word in a story that he selected, we only looked at encounters of the word on a later day as possible evidence of learning. So improved performance in the student selected story cannot be the cause of these results. Second, this study took place over an academic year (from September 2003 through May 2004), so the results are not a simple novelty effect.

Another possibility is that the results are not due to higher student motivation or engagement, but instead are due to students choosing stories that were more appropriate pedagogically. However, such is not the case. When it was the Reading Tutor's turn to pick a story, it selected a story at the student's estimated reading level that he had not previously read. Students could select stories at any level of difficulty and could reread stories. Students and the Reading Tutor selected stories at nearly the same level of difficulty. For the 93347 student-chosen stories in which they spent at least one minute, the mean difficulty was 2.27 (where 2 is second grade reading level and 3 is third grade reading level). For the 100417 tutor-chosen stories in which the student spent at least one minute, the mean difficulty was 2.01. So the difficulty levels are comparable across selection types. Furthermore, the student ability to reread stories likely hurt their learning, since we have found that rereading previously read stories leads to less efficient learning [8].

Given that students learn more when they select which story to read, an obvious question is whether the benefit outweighs the cost. As a simple model, we consider how much time students spent reading stories (presumably productive learning time) vs. how much time they spent picking stories (presumably non-productive learning time—at least for learning how to decode words). In this data set, students read for 3053 hours and spent 503 hours selecting which story to read. If the tutor instead chose a story and forced the student to read it, those 503 hours could have instead been spent reading, resulting in 3053 + 503 = 3556 hours of instruction. Unfortunately, since stories selected by the tutor only result in 80% as much learning, the amount of comparable time on task would be 3556 * 0.8 = 2845 hours—fewer than the 3053 hours that students actually spent reading. So it appears that allowing student to select activities is worth the time cost. Actually, this analysis is an underestimate of the benefit since there are probably benefits to persistence (how long students are willing to use the tutor) that are unaccounted for. Also, as anyone who has observed actual field conditions for computer tutors can attest, the part about forcing students to read a story is problematic, at best, to instantiate.

3.1 Median as a Measure of Central Tendency

There was considerable variance as the per-student estimates of B ranged from -201 to 2185. Since some students have relatively few observations, their estimated value of B is rather inaccurate. One approach would be to trim or cap outliers and then take the mean. However, this approach requires first determining what constitutes an outlier. Presumably the student who the model claims learned 2000 times as much is a result of a poorly estimated parameter value and should be ignored, but what of the students who learned 5 times as much as a result of tutor control? 3 times as much? Triple the rate of learning is implausible, but possible. Rather than create an arbitrary threshold for when a student is an outlier, we simply use the median as a measure of central tendency. This process compresses the effects of outliers. For example, the median would be the same if the student whose B parameter was 2000 was instead 2. In

contrast, the mean is sensitive to magnitude, and in fact the mean value of *B* is an implausible 20.2.

There is a second difficulty with using the mean. Imagine a population consisting of three students. One student learns three times as much when the tutor picks the story (*B*=3) while the two other students learns one-third as much when the tutor picks the story (*B*=0.33). The mean of this sample is (3 + 0.33 + 0.33) / 3 = 1.22—a counter intuitive result given that 3 and one-third are exactly opposite effects, and twice as many students are having negative results. Therefore, we feel the use of median as a measure of central tendency is appropriate in this context.

3.2 Which Students Benefit from Learner Control?

We have established that students in the Reading Tutor benefit from a degree of learner control, and that this control is worth the time it takes away from learning. The final question we address is whether all students benefit from learner control? If certain students do not benefit from or need the motivational effect of learning control, we can make their learning more effective by having the tutor select materials for them.

We present two approaches for finding such students. The first approach is top-down and disaggregates the results by the student's grade level. We found that the 130 first grade students (typically 6 years of age) had a median *B* of 0.98, the 134 second graders a 0.89, and the 70 third graders a 0.49. We had only 8 and 4 students in grades four and five, respectively, so the medians were not reliable. This trend suggests that younger readers are insensitive to the material they are reading. As readers become more advanced, they develop preferences in what they wish to read and they are less engaged in tutor selected stories.

The second approach is bottom-up, and instead treats students as individuals and attempts to discover similarities between students who benefit from learner control vs. those who do not. Our approach is to use the student's estimated *B* score as a noisy category label and treat the task as one of classification. If a student's *B* score was below 0.9, this student was labeled as benefiting from learner control (190 students). If the student's *B* score was above 1.1, the student was labeled as benefiting from tutor control (133 students). The remaining 23 students were not used for the training. For our classifier, we used logistic regression with the following independent variables:

- The student's grade (covariate)
- The student's grade relative Woodcock Reading Mastery Test composite score [9] (covariate)
- The student's gender (factor)
- Whether the student was receiving learning support services (factor) which took on three levels: yes (32 students), no (169 students), and unknown (145 students)

We chose those dependent variables since they are likely to be known by teachers and researchers at the beginning of the school year. Only the second variable, the grade relative Woodcock Reading Mastery Test score, is difficult to obtain, and we suspect that most standardized test scores for the domain would do just as well.

The model's Nagelkerke R^2 was a rather poor 0.045, with the only statistically reliable independent being the student's gender. Apparently girls were more likely to benefit from tutor control, while boys are more likely to benefit from learner control.

One possible explanation is that students tend to select stories that are less than optimal from a learning perspective. If the Reading Tutor is selecting pedagogically better stories than the student, then students who are engaged regardless of the material being read will benefit more from tutor control. Perhaps young girls are more interested in reading than young boys? In fact, we found that girls scored reliably higher ($P<=0.001$, two-tailed t-test) on measures of both academic and recreational reading interest [10].

One explanation for the low model fit is the noisy class labels. The per-student estimates of B are individually suspect, and we are relying on the fact that we have hundreds of estimates to try to derive a pattern. That the model is doing a better job at separating students than the fit statistics suggest is evident from looking at its predictions. Students the model predicted would benefit from learner control (N=261) had a median B parameter of 0.66, students the model predicted would be hurt by learner control (N=61) had a median B of 1.23—a wide degree of separation.

4 Contributions, Future Work, and Conclusions

This paper's main contributions is measuring the effect that perceived learner control in an ITS has on student learning. Prior work, such as [6] examined the effects of learner customization and found that fairly minor modifications (icons used to represent the student, referring to friends of the student in background text, etc.) resulted in statistically reliable gains in learning. However, this study was of short duration: only four experimental sessions. It is unclear whether cosmetic modifications will succeed in maintaining student interest over the course of an entire academic year. Our study shows that a fairly minor choice of which story to read next results in noticeable differences in the student learning of the words in the story. Our study also determines for whom such a benefit is more likely to occur. Namely, that boys are more likely to benefit from the Reading Tutor's perceived control than girls.

This work also introduces a methodology, learning decomposition, for investigating issues involving the student's affective state. This approach required no additional data collection. In fact, this analysis was not conceived of until roughly two years after the study ended in Spring 2004. The approach of learning decomposition is broadly applicable, and it is not simply appropriate for studying affective factors such as engagement or motivation. Our prior work with this technique [8] investigated the impact of cognitive factors (spaced practice) and suspected reading curriculum issues (rereading the same story vs. reading new stories).

Although this work makes a good start in creating a framework for investigating effects of learner control, it is only a start. First, this analysis has not been replicated via a classic controlled study. Our prior belief was that the difference in learning gains would be small, and a controlled study unwarranted. A 25% increase in learning is substantial, and worth exploring further. Second, our analysis only pertains to a specific instantiation of selecting the next task. Would constraining the space of learner decisions still result in the same benefit? For example, imagine if we presented many fewer stories for the student to choose among. This design change would reduce the time it takes students to select stories, resulting in more time on task. But would we get the same increase in learning as we found in this study?

One other limiting factor is scope. We have analyzed one type of learner decision, what story (i.e. problem) to work on next. Do other types of learner control in ITS show similar benefits? How much learner control is needed? Is one choice at the

beginning of the session with the tutor sufficient, or do students require periodic choice points? There is a broad scope of possible instantiations of learner control, of which this paper examines only one. One final limitation is that learning decomposition requires fine-grained markers of student progress such as performance data on individual problems.

There is also a limitation that not all students are engaged and on-task when the tutor selects stories. For example, students finished 47% of stories that they selected vs. only 38% of stories selected by the tutor—however, we used all of the words the student read, even those in uncompleted stories, to train our model. Presumably this differential attrition was from students who were not motivated by the tutor's choice. Therefore, the estimate of learning in tutor-chosen stories is from a group of students who were generally intrinsically more motivated to read than in student-chosen stories. We do not see a good way to address this issue. However, this bias causes tutor-chosen stories to look more effective for learning than they actually are, which would weaken the main result in the paper that student-chosen stories result in more learning. Thus the real difference is likely stronger than what we report.

In conclusion, we have shown that student choice—at least for our instantiation of it in the Reading Tutor—not only increases student motivation, it also increases student learning for activities the student chose to perform vs. those he did not select. We have also found that older children and boys are more sensitive to learner control.

Acknowledgements. This work was supported by the National Science Foundation, ITR/IERI Grant No. REC-0326153. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the National Science Foundation

References

1. Mostow, J. and G. Aist, *Evaluating tutors that listen: An overview of Project LISTEN*, in *Smart Machines in Education*, K. Forbus and P. Feltovich, Editors. 2001, MIT/AAAI Press: Menlo Park, CA. p. 169-234.
2. Heiner, C., J.E. Beck, and J. Mostow. *Improving the help selection policy in a Reading Tutor that listens*. in *Proceedings of the InSTIL/ICALL Symposium on NLP and Speech Technologies in Advanced Language Learning Systems*. 2004. p. 195-198 Venice, Italy.
3. Aleven, V. and K.R. Koedinger. *Limitations of Student Control: Do Student Know when they need help?* in *Proceedings of the 5th International Conference on Intelligent Tutoring Systems*. 2000. p. 292-303 Montreal: Berlin: Springer Verlag.
4. Beck, J. *Engagement tracing: using response times to model student disengagement*. in *Proceedings of the 12th International Conference on Artificial Intelligence in Education (AIED 2005)*. 2005. p. 88-95 Amsterdam: IOS Press.
5. Becker, D.A.A. and M.M. Dwyer, *Using hypermedia to provide learner control.* Journal of Educational Multimedia and Hypermedia, 1994. **3**(2): p. 155-172.
6. Cordova, D.I. and M.R. Lepper, *Intrinsic motivation and the process of learning: Beneficial effects of contextualization, personalization, and choice.* Journal of Educational Psychology, 1996. **88**(4): p. 715-730.
7. Aist, G. and J. Mostow, *Faster, better task choice in a reading tutor that listens*, in *The Path of Speech Technologies in CALL: Foundations, Prototypes, and Evaluation*, V.M. Holland and F.N. Fisher, Editors. in press, Routledge Publishers: New York.
8. Beck, J. *Using learning decomposition to analyze student fluency development*. in *ITS2006 Educational Data Mining Workshop*. 2006. p. Jhongli, Taiwan.
9. Woodcock, R.W., *Woodcock Reading Mastery Tests - Revised (WRMT-R/NU)*. 1998, Circle Pines, Minnesota: American Guidance Service.
10. McKenna, M.C., D.J. Kear, and R.A. Ellsworth, *Children's attitudes toward reading: a national survey.* Reading Research Quarterly, 1995. **30**: p. 934-956.

Emotion and Affect

This page intentionally left blank

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Affect and Usage Choices in Simulation Problem-Solving Environments

Ma. Mercedes T. RODRIGO[1], Ryan S.J.d. BAKER[2], Maria C.V. LAGUD[3], Sheryl A.L. LIM[1], Alexis F. MACAPANPAN[3], Sheila A.M.S. PASCUA[3], Jerry Q. SANTILLANO[1], Leima R.S. SEVILLA[3], Jessica O. SUGAY[1], Sinath TEP[1] and Norma J.B. VIEHLAND[3]

[1] Department of Information Systems and Computer Science, Ateneo de Manila University
[2] Learning Sciences Research Institute, University of Nottingham
[3] Education Department, Ateneo de Manila University
mrodrigo@ateneo.edu, ryan@educationaldatamining.org

Abstract. We investigate the relationship between a student's affect and how he or she chooses to use a simulation problem-solving environment, using quantitative field observations. Within the environment studied, many students were observed gaming the system (cf. Baker et al, 2004), while few students engaged in off-task behavior. We analyze which affective states co-occur with gaming the system, and which affective states precede gaming behavior. Boredom and confusion appear both to precede gaming behavior and to co-occur with gaming behavior; delight and flow are negatively associated with gaming behavior.

Introduction

In recent years, there has been increasing interest in understanding how a student's attitudes and affective state concretely alters his or her behavior, as he or she uses an interactive learning environment [1,5,6,8,16]. Much of this research was conducted by assessing each student's general affect and/or attitudes towards the learning environment through questionnaires given before or after system usage. Then, data mining is used to link each student's behavior to his or her self-reported attitudes and affect [1,5,6,16]. Studies following this approach have produced understanding of the links between affect, attitudes, and students' actions in learning systems. The majority of this work, however, has only studied the links between a student's generalized affect towards a system and his or her behavior, rather than the relationship between a student's affective states at a specific moment, and his or her behavior at the same time or immediately afterwards. (One exception, [8], explicitly models student affect at specific times, but represents affect solely as an outcome of the interaction between the student and the system, rather than as a factor which potentially alters student behavior).

Studying affect only in general, rather than at specific times, limits the questions that an analysis of the relationship between affect and behavior can address. To give a

pair of examples: Even if we have evidence that gaming the system is associated with frustration [cf. 16], or that taking a long time between answering attempts is generally linked with fear of making errors [cf. 1], does that mean that a student first experiences the affective state (frustration, fear), and then engages in the behavior? Or does the student experience the affective state as he or she is engaging in the behavior? Or is the relationship between affect and behavior more complex still? Questionnaire assessments of affect and attitudes are an important beginning towards showing the relationship between affective states and behavior in learning systems, but leave some important questions unanswered.

In this paper, we analyze the relationship between a student's affective state, at a given time, and their behavior, both at that time and shortly thereafter. In this manner, we can study more precisely how specific affective states are antecedents to and/or co-occur with a student's choices of how to interact with a learning system.

More specifically, we will study two categories of behavior found to be associated with poorer learning – gaming the system [cf. 4], and off-task behavior [2]. Prior research has shown that gaming the system is generally associated with frustration [16], but it is not yet clear whether students experience frustration while gaming or whether students experience frustration and then game the system shortly afterwards. Additionally, both gaming the system and off-task behavior have been found to be associated with lack of interest in the system's subject matter [2, 16] – hence, boredom may also be associated with gaming the system and off-task behavior.

In this paper, we study the relationship between affect and student usage choices within a simulation problem solving environment designed to be both fun and educational, The Incredible Machine: Even More Contraptions [15]. Studying this issue within the context of a simulation problem solving environment enables us both to learn about the relationships between specific behaviors and affective states, and to learn how the frequencies of behaviors and affective states differ between simulation learning environments and the other types of environments where these behaviors and affective states have been studied [cf. 4, 9, 17]. In particular, it has been found that off-task behavior is considerably less common in action games than in intelligent tutoring systems [cf. 4, 17]. By seeing whether the incidence of off-task behavior in a simulation learning environment designed to be both fun and educational is more similar to the incidence of off-task behavior in action games – systems designed primarily with the goal of fun – or intelligent tutoring systems – systems designed primarily with the goal of learning, we can understand better how students view learning environments designed with both goals in mind.

1. Methods

The relationship between affective states and usage choices was studied within a high school mathematics class in a private school in urban Manila, in the Philippines. Student ages ranged from 14 to 19, with an average age of 16. Thirty-six students participated in this study (17 female, 19 male).

Each student used The Incredible Machine: Even More Contraptions [15] (shown in Figure 1), a simulation environment where the user completes a series of logical "Rube Goldberg" puzzles. In each puzzle, the student has a pre-selected set of

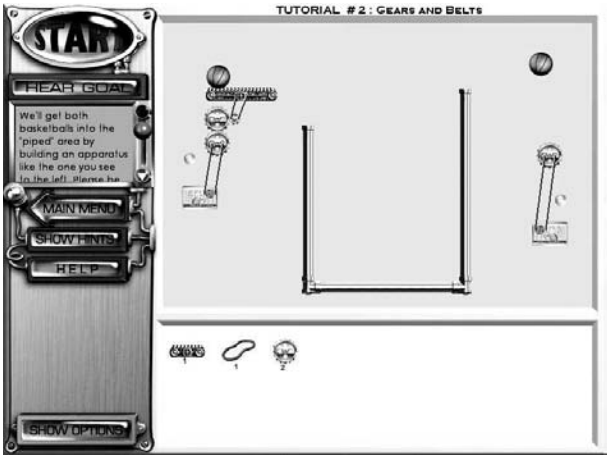


Figure 1. A screen shot from The Incredible Machine: Even More Contraptions.

objects to use, such as scissors, ropes, and pulleys, electrical generators, and animals. The student must combine these objects in order to accomplish a pre-defined goal, such as lighting a candle or making a mouse run. If a student is stuck, he or she can ask for a hint; hint messages display where items should be located in a correct solution to the current problem (but do not show which item should be placed in each location).

Each student used The Incredible Machine for ten minutes, and each student's behavior and affect was observed several times as he or she used The Incredible Machine. The observations were conducted using a method which incorporated aspects of Baker et al's [4] quantitative field observations of student behavior categories, and Craig et al's [9] laboratory observations of affect. The observations were carried out by a team of six observers, working in pairs. As in Baker et al, each observation lasted twenty seconds, and was conducted using peripheral vision in order to make it less clear exactly when an observation was occurring. If two distinct behaviors were seen during an observation, only the first behavior observed was coded, and any behavior by a student other than the student currently being observed was not coded.

It was not possible for the entire class to use the software at the same time, due to the size of the school computer laboratory; hence, students used the software in groups of nine (one student per computer), during their class time. Each pair of observers was assigned to three students and alternated between them. Since each observation lasted twenty seconds, each student was observed once per minute. Observing students more frequently than in [4] or [9] made it possible to directly analyze the relationship between a student's affective state at a given time and their usage choices shortly thereafter.

Within an observation, each observer coded both the student's behavior and affective state, using coding schemes developed in prior research. The observers trained for the task through a series of pre-observation discussions on the meaning of the usage and affective categories.

The usage categories coded were adapted from [4], and are as follows:

1. **On-task** – working within The Incredible Machine
2. **On-task conversation** – talking to the teacher or another student about The Incredible Machine, or its puzzles

3. **Off-task conversation** – talking about any other subject
4. **Off-task solitary behavior** – any behavior that did not involve The Incredible Machine or another individual (such as reading a magazine or surfing the web)
5. **Inactivity –** instead of interacting with other students or the software, the student stares into space or puts his/her head down on the desk.
6. **Gaming the System** – sustained and/or systematic guessing, such arranging objects haphazardly or trying an object in every conceivable place. Also, repeatedly and rapidly requesting help in order to iterate to a solution.

The affective categories coded were drawn from [11]. Since many behaviors can correspond to an emotion, the observers looked for students' gestures, verbalizations, and other types of expressions rather than attempting to explicitly define each category. The categories coded were:

1. **Boredom** – behaviors such as slouching, and resting the chin on his/her palm; statements such as "Can we do something else?" and "This is boring!"

2. **Confusion –** behaviors such as scratching his/her head, repeatedly looking at the same interface elements; statements such as "I don't understand?" and "Why didn't it work?"

3. **Delight** – behaviors such as clapping hands or laughing with pleasure; statements such as "Yes!" or "I got it!"

4. **Surprise** – behaviors such as jerking back suddenly or gasping; statements such as "Huh?" or "Oh, no!"

5. **Frustration –** behaviors such as banging on the keyboard or pulling at his/her hair; statements such as "This is annoying!" or "What's going on?!?"

6. **Flow** – complete immersion and focus upon the system [cf. 10]; behaviors such as leaning towards the computer or mouthing solutions to him/herself while solving a problem

7. The **Neutral** state, which was coded when the student did not appear to be displaying any of the affective states above, or the student's affect could not be determined for certain.

Some of these affective categories may not be mutually exclusive (such as frustration and confusion), though others clearly are (delight and frustration). For tractability, however, the observers only coded one affective state per observation.

Past research has suggested that brief observations can be reliable indicators of a student's affective state, whether carried out live [11] or by watching screen-capture videos [12]. 706 observations were collected, for an average of 19.6 observations per student. Inter-rater reliability was acceptably high across all observations – Cohen's [7] κ=0.71 for usage observations, κ=0.63 for observations of affective state.

2. Results

2.1 Overall Results

The two most common behavioral categories observed were working on-task with the software (80% of observations), and talking on-task (9% of observations). The combined frequency of the on-task categories was higher than is common in traditional classrooms [13,14] or intelligent-tutor classrooms [4], though lower than the frequency seen among students playing non-educational action games [17].

Gaming the system was the third most common category of behavior, observed 8% of the time. Although the overall frequency of gaming was higher than in previous observational studies [cf. 4], the percentage of students ever seen gaming – 36% – was within the range observed in previous studies. Both off-task conversation and off-task solitary behavior were quite rare, occurring 0.5% and 0.3% of the time – this frequency is considerably lower than the frequency of off-task behavior in intelligent-tutor classrooms [4] but comparable to the frequency of off-task behavior among students playing non-educational action games [17]. Therefore, one potential interpretation of this finding is that students are generally less likely to go off-task when using environments designed with fun as a primary goal. Another possibility is that our observation technique inhibited students' willingness to be visibly off-task, although this hypothesis does not explain why students engaged in relatively high amounts of gaming the system. And, of course, it is also possible that differences in the population studied (such as age, culture, and type of school) from populations in earlier studies may explain the rarity of off-task behavior in our sample.

The most common affective state observed was flow, coded in 61% of the observations. The dominance of the flow state is similar to results seen in prior studies of affect in students using intelligent tutoring systems [9]. The second most common category was confusion, observed 11% of the time. Boredom (7%), frustration (7%), delight (6%), and the neutral state (5%) were each seen in a small but definite proportion of the observations. Boredom was relatively less common than in previous work studying affect in intelligent tutoring systems (Craig et al, 2004), but frustration was relatively more common. Surprise was the rarest category, but was still observed (3%).

2.2 Affect and the choice to game the system

In this section, we will discuss the relationship between a student's affective state and behavior. We focus on gaming behavior, because gaming the system is known to be associated with poorer learning [4,5,16] and also because gaming was observed with reasonably high frequency in our observations, making statistical inference feasible. We study the relationship between gaming behavior and affective states in two fashions. First, we will study what affective state a student experiences at the time he or she is gaming the system. Second, we will study the antecedents of gaming by investigating what affective states students experience one minute before gaming.

At the exact time a student is gaming, he or she is most frequently bored (39% of the time). Since students are generally bored 7% of the time, they are over 5 times more likely to be bored when they are gaming, a significant difference, $\chi^2(1, N=706) =$

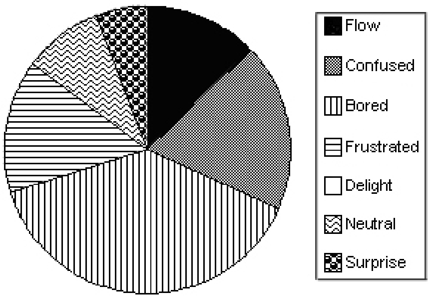


Figure 2. The frequency of affective categories, at the time a student is gaming the system.

95.02, $p<0.001$. The second most common affective state among a student as he or she games is confusion (19% of the time). Students are generally confused 11% of the time; the difference in frequency between gamers and non-gamers is marginally significant, $\chi^2(1, N=706)=3.01$, $p=0.08$. The third most common affective state among a student as he or she games is frustration (15% of the time). Since students are generally frustrated 7% of the time, they are more than twice as frequently frustrated when they are gaming, a significant difference, $\chi^2(1, N=706)=5.93$, $p=0.01$. Flow, observed only 13% of the time among gaming students (61% of the time overall), is much less common when a student is gaming, $\chi^2(1,N=706)=57.68$, $p<0.001$. The neutral state (9%) and surprise (6%) were not significantly related to gaming. Delight and gaming were never seen in conjunction, in any observation. The overall frequency of each affective state at the time of gaming is shown in Figure 2.

Another way to analyze the relationship between affective states and gaming is to investigate which affective states serve as antecedents to gaming. We can determine what affective states are antecedents to gaming by looking at the probability of gaming one minute after an affective state is observed.

According to our data, if a student is bored, he or she games one minute later 21% of the time, over twice the overall average frequency of gaming, $\chi^2(1, N=634)=10.45$, $p<0.01$. If a student is confused, he or she games one minute later 17% of the time, twice the overall average frequency of gaming, $\chi^2(1, N=634)=7.73$, $p<0.01$.

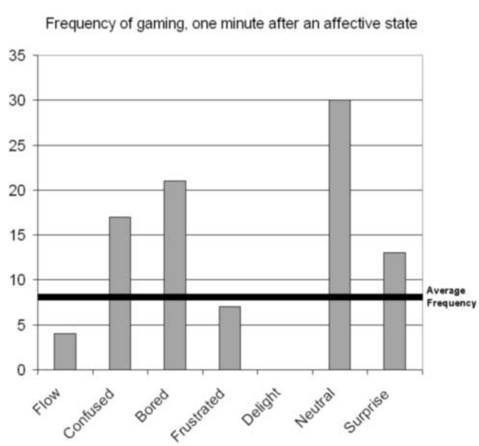


Figure 3. The frequency of gaming one minute after being in a specific affective state.

In addition, if a student is neutral, he or she games one minute later 30% of the time, almost four times more than the overall average, $\chi^2(1, N=634)=22.58$, $p<0.001$.

However, if a student is frustrated, he or she games one minute later only 7% of the time, not significantly different than the overall average frequency of gaming, $\chi^2(1, N=634)=0.05$, $p=0.83$. If a student is surprised, he or she games one minute later 13% of the time, also not significantly different than the overall average frequency of gaming, $\chi^2(1, N=634)=0.40$, $p=0.53$.

If a student is in flow, he or she games one minute later only 2% of the time, much lower than the overall average frequency of gaming, $\chi^2(1, N=634)=22.89$, $p<0.001$. Finally, if a student is delighted, he or she never games one minute later, within our data. The frequency of gaming behavior, one minute after each affective state, is shown in Figure 3.

Hence, boredom and confusion both co-occur with gaming, and are antecedents to gaming the system. Curiously, however, frustration does not appear to be an antecedent to gaming, although it co-occurs with gaming. Additionally, delight and flow are negatively associated with gaming the system, both at the same time and one minute later.

Unexpectedly, the neutral affective state also appears to be an antecedent to gaming. This somewhat surprising finding may suggest that an affective state we did not include in our coding scheme may also be related to gaming (and that this state was coded as neutral by the observers, for lack of a better way to code it.)

3. Discussion and Conclusions

In this study, we have studied student usage choices and affective states in a simulation problem solving environment, The Incredible Machine. One finding is that off-task behavior is quite rare in The Incredible Machine, as in action games [17] but in contrast to intelligent tutoring systems [4]. At the same time, gaming the system is at least as common in The Incredible Machine as it is in intelligent tutors. A potentially interesting question for future work is whether students view gaming the system in the same fashion within game-like simulation learning environments as in intelligent tutoring systems. Students appear to know that gaming the system is inappropriate within tutoring systems, since they hide their gaming behavior from their teachers. Do they believe that gaming the system is inappropriate behavior within more game-like environments? Gaming the system is often not considered appropriate behavior within games (as shown by the lack of respect some game-players have for over-use of cheats and hints within games), but there is also the sense that since games are primarily for fun, it is acceptable to use them in any fashion. If students transfer a sense that gaming the system is appropriate behavior from pure-entertainment games to educational games, there may be implications for how educational games should respond when students game the system.

In terms of the relationship between gaming the system and affective states, we find that boredom and confusion both co-occur with gaming behavior and serve as antecedents to it. Frustration co-occurs with gaming, but does not appear to be an antecedent to gaming. This raises some questions about how frustration and gaming are related – does frustration lead to gaming, but too rapidly for observations spaced a minute apart to detect it? Or do students become frustrated when they choose to game the system and still do not immediately obtain the correct answer? Another affective state, the neutral state, also appears to be a strong antecedent to gaming – we currently

have no definitive explanation why. In both cases further research, through more fine-grained video analysis or interviews, may help us understand these patterns better.

In general, knowing that boredom and confusion are antecedents to gaming the system suggests that detecting boredom and confusion [cf. 11] might signal a new way for adaptive systems to respond to gaming behavior. A system that knew a student was bored or confused, and that the student had a history of gaming behavior, could offer the type of supplementary support shown to help gaming students learn better [cf. 3] before the student even starts to consider gaming the system.

Acknowledgements

We would like to thank Anita Rilloraza and the Kostka School of Quezon City for assistance in conducting the studies at the schools, and Genaro Rebolledo-Mendez and Ulises Xolocotzin for useful suggestions and feedback. This work was funded in part by a fellowship from the Learning Sciences Research Institute.

References

[1] Arroyo, I., Woolf, B. (2005) Inferring learning and attitudes from a Bayesian Network of log file data. *Proceedings of the 12th International Conference on Artificial Intelligence and Education*, 33-40.

[2] Baker, R.S.J.d. (in press) Modeling and Understanding Students' Off-Task Behavior in Intelligent Tutoring Systems. To appear in *Proceedings of ACM SIGCHI: Computer-Human Interaction 2007.*

[3] Baker, R.S.J.d., Corbett, A.T., Koedinger, K.R., Evenson, E., Roll, I., Wagner, A.Z., Naim, M., Raspat, J., Baker, D.J., Beck, J. (2006) Adapting to When Students Game an Intelligent Tutoring System. *Proceedings of the 8th International Conference on Intelligent Tutoring Systems*, 392-401.

[4] Baker, R.S., Corbett, A.T., Koedinger, K.R., and Wagner, A.Z. (2004) Off-Task Behavior in the Cognitive Tutor Classroom: When Students "Game the System". Proceedings of ACM CHI 2004: Computer-Human Interaction, 383-390.

[5] Baker, R.S., Roll, I., Corbett, A.T., Koedinger, K.R. (2005) Do Performance Goals Lead Students to Game the System? *Proceedings of the International Conference on Artificial Intelligence and Education (AIED2005)*, 57-64.

[6] Beal, C.R., Qu, L., Lee, H. (2006) Classifying learner engagement through integration of multiple data sources. *Paper presented at the 21st National Conference on Artificial Intelligence.*

[7] Cohen, J. (1960) A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement,* 20, 37-46.

[8] Conati, C., Maclaren, H. (2004) Evaluating a probabilistic model of student affect. *Proceedings of the 7th International Conference on Intelligent Tutoring Systems,* 55-66.

[9] Craig, S.D., Graesser, A.C., Sullins, J., Gholson, B. (2004) Affect and learning: an exploratory look into the role of affect in learning with AutoTutor. *Journal of Educational Media,* 29 (3), 241-250.

[10] Csikszentmihalyi, M. (1990) *Flow: The Psychology of Optimal Experience.* New York: Harper and Row.

[11] D'Mello, S. K., Craig, S. D., Witherspoon, A., McDaniel, B., Graesser, A. (2005). Integrating affect sensors in an intelligent tutoring system. In "Affective Interactions: The Computer in the Affective Loop Workshop" In conjunction with *International conference on Intelligent User Interfaces,* 7-13.

[12] de Vicente, A., Pain, H. (2002) Informing the detection of the students' motivational state: an empirical study. *Proceedings of the 6th International Conference on Intelligent Tutoring Systems*, 933-943.

[13] Lee, S.W., Kelly, K.E., Nyre, J.E. Preliminary Report on the Relation of Students' On-Task Behavior With Completion of School Work. *Psychological Reports,* 84 (1999), 267-272.

[14] Lloyd, J.W., Loper, A.B. Measurement and Evaluation of Task-Related Learning Behavior: Attention to Task and Metacognition. *School Psychology Review,* 15 (3) (1986), 336-345.

[15] Sierra Online, Inc. (2001) *The Incredible Machine: Even More Contraptions.*

[16] Walonoski, J.A., Heffernan, N.T. (2006) Detection and Analysis of Off-Task Gaming Behavior in Intelligent Tutoring Systems. *Proceedings of the 8th International Conference on Intelligent Tutoring Systems,* 382-391.

[17] Ziemek, T. (2006) Two-D or not Two-D: gender implications of visual cognition in electronic games. *Proceedings of the 2006 Symposium on Interactive 3D graphics and games,*183-190.

Getting Under the Skin of Learners: Tools for Evaluating Emotional Experience

Madeline ALSMEYER[1], Rose LUCKIN[2], Judith GOOD[1]
[1]*IDEAS lab, School of Science and Technology, University of Sussex, UK*
[2]*London Knowledge Lab, Institute of Education, University of London, UK*

Abstract: This paper reports a study that evaluated a methodology and tool for eliciting the emotional experience of language learning. It has looked at the relative rate of change of different emotions over the course of a lesson and also the differences between self-report and observer reports of changes in emotion. Implications for the design of affectively intelligent systems are discussed.

Keywords: Emotion and affect, language learning, classroom-based experimentation, qualitative, case study, think aloud

1. Introduction

Successful language learning depends on the willingness of the learner to interact with other members of a learning group, or a community of native speakers. Learning which depends on such sociality is undoubtedly governed by a learner's affective state. The work reported in this paper is concerned with clarifying the affective needs of such learners, in support of the design of affectively intelligent systems [1-5].

Our research is concerned with how a learner's affective state is related to the context in which learning occurs. Can a particular context be enhanced to support the emotional experience of learning? Are there ways in which learners (and others in the context) can adapt the learning context to better support their affective needs?

This research focus is grounded in sociocultural theories of learning [6, 7] and a belief that learning within an individual is the result of that individual's interactions with others within a particular cultural context. Learning is therefore dependent upon the nature of the context and what it affords learners in terms of physical tools, people and the location itself [8, 9]. If we accept that a learner's affective state (at least, in part), is also dependent upon her social interactions then both cognitive and affective states can be influenced by what a context is able to facilitate.

To evaluate the effectiveness of an affective learning environment, reliable data about the emotional experiences of the participants are needed. Such data can take the form of biophysical sensor recordings, interaction logs [10] and self-report [11].

The aim of this study was to evaluate a methodology and a set of tools for eliciting self-reports of emotional experience. This paper begins with an exploration of the tools and methodologies currently adopted in analysing students' emotional experiences. This is followed by a description of the tool and methodology developed by the authors of this paper. Lastly we describe a study carried out to evaluate the tool and methodology and present the preliminary results of this study.

2. Measuring Emotional Experience

Relatively few studies have attempted to uncover students' emotional experiences of learning. Those that have [12-16] have made two clear methodological decisions. The first relates to when to collect a participant's self-report: throughout the academic experience, directly afterwards or using stimulated recall, i.e, using a device to provoke recollection of an event? The second relates to the appropriate method for collecting self-reports. There are three main methods: free response approach, the discrete emotion response approach and the dimensional emotion response approach.

2.1. Free Response Approach

This approach asks participants to describe how they felt during an academic experience in their own words. The verbalisations are then clarified through participant-researcher discussion until a consensus is reached.

This approach allows participants to describe their emotional experiences in a personally meaningful way, and is generally believed to result in specific and accurate self reports of emotional experience [11]. It relies however on participants being comfortable and capable of describing their emotions and upon the researcher's skills in conducting the interviews. Furthermore, any clarification discussion with the participants requires them to feel comfortable enough to defend or elaborate on their verbalisation. This may render the technique more appropriate for more mature participants, or those with high emotional intelligence [17].

2.2. Dimensional Emotions Approach

This approach [18] attempts to build a structured description of an emotion within a dimensional space, using axes (which the researchers believe adequately differentiate one emotion from all others). Such studies require participants to locate their emotion in the space constructed by the two or three axes.

This approach revolutionised the way in which researchers talk about emotion, but it suffers from a number of limitations. For example, describing an emotional experience using dimension axes such as valence, arousal and tension is an abstract task, as participants are unlikely to break down their own emotional experiences in this way when they are reflecting on them personally. Furthermore, it is difficult to judge whether participants who chose the same dimensional space actually felt the same emotion, as it is possible, for two completely separate emotions to occupy roughly the same dimensional space, (for example, fear and anger share the same region of a two-dimensional space, constructed by the axes valence and arousal) [19].

2.3. Discrete Emotions Approach

This approach requires participants to describe the way they are feeling using a given list of words. Participants either rate their experienced emotions along a scale, e.g. a Likert-style scale or a verbal scale (strongly – not at all), or respond to questions or statements, which summarise a particular emotional state without directly mentioning the emotions. Methods vary from participants reporting only their most pertinent emotions, to reporting a degree of strength for all the emotions listed by the researcher.

This method suffers from various drawbacks, for example, by specifying only a limited set of words, the researcher may be restricting what the participant can report as experienced with a resultant loss of data. Furthermore, different studies using this method have tended to use emotional descriptive words which are applicable to their study or participant group, meaning that results cannot be compared across studies. Lastly, using a set of descriptive emotional words may prompt the participant to remember something more about their experience, but may also lead the participant to falsely report an emotion in order to please the researcher.

3. An Evaluative Emotion Reporting Tool and Methodology

The tools and methodologies described in this paper are intended to streamline the process of collecting self-reports of emotional experience within a real world educational context. The tools developed must be able to collect data quickly, easily and as reliably as possible. Stimulated recall was chosen as it allows self-reports on emotional experience to be collected in a way which does not interfere with the participants' learning within lessons, and it provides flexibility with respect to when the subsequent interviews can be carried out.

In conjunction with the stimulated recall approach, the study used a discrete emotions approach to structure the collection of self-reports. This approach was chosen since the participants' age range (13 – 18 years) meant the participants might lack the vocabulary or confidence necessary to adequately report their emotional experience using a free-response approach. It was also our concern that the level of abstraction required for the dimensional approach would be unsuitable for our participants.

We view this study as a starting point of a user centred design process for the development of an intervening technology. As such, we need to have an understanding of the types of emotional language and labels that are appropriate for use within this context. Those adopted for use in this study were based on Pekrun's work [13] which found that the common emotions experienced during academic interactions were: enjoyment, hope, pride, relief, anger, anxiety, shame and boredom.

The first phase of the tool aims to analyse the emotional state of the learner at the beginning of the learning experience, since without this knowledge it is difficult to contextualise the resulting reported experience. The tool is made up of a number of sliders which are moveable along an axis of "not at all X" where X is the emotion being evaluated to "extremely X". We chose to use semantically labelled sliders, since results from pilot studies suggested participants found Likert scales quite restrictive.

The second phase of the tool is used during the stimulated recall interview to collect each participant's self report of emotional experience. The tool simply asks the learner to report how they remember feeling at a certain point in their academic experience in comparison with how they remember feeling at the beginning of the experience by selecting either "More X" (where X is the emotion currently being investigated), "The same" or "Less X". Qualitative terms have been used since we are predominantly interested in how the emotional experience *changes* with time, and collecting this information doesn't necessitate we know the exact quantities of change, just the direction of change. Furthermore, the interviews generally took place between 1 day and a week after the learning experience itself, we therefore believed it would be less suitable to ask our participants for precise quantities of change.

4. Methodology

4.1. The Study

The aim of this short study was to begin the evaluation of an emotion reporting tool and methodology, based on how easy the tool was for the participants to use, how flexibly the tool could fit into a real learning context and finally, how reliable the tool was in terms of collecting participants' emotional experiences of learning.

4.2. Participants

The study was carried out in a British school, with 3 separate German language classes in two separate school years, year 9 (13 – 14 year olds) and year 12 (16 – 17 year olds). A total of 22 students in 11 pairs took part. 7 pairs of students were selected from the year 9 classes to participate in the study. The year 12 class is significantly smaller in size which meant all 8 students (i.e. 4 pairs) within the class took part.

4.3. Procedure

The study took place over the space of a month in the autumn/winter term. One pair of students were video recorded per classroom session. Typically the pairs of students chosen were classroom neighbours, or shared the same table. Each member of the pair was required to complete the first part of the emotion reporter tool at the beginning of the classroom session. Throughout the lesson, two pieces of footage were recorded for each student. One recording was made as they participated in a verbal activity (generally with the teacher) in front of the whole class and the second recording was made as they participated in a small group activity. The participants completed the stimulated recall interview individually within one week of the classroom activity. During this interview the participants were shown the clips of themselves participating in class. Each video clip was paused once every 30 seconds at which point the participant was asked to complete phase 2 of the emotion reporting tool (I.E reporting for each emotion whether they recalled feeling more X, less X or the same X, where X is the emotion being reported on). After reporting on their emotional experience, a semi-structured interview was conducted in order to ascertain whether any emotions had been experienced by the students which hadn't been enquired about in the emotion reporting tool and in order to capture thoughts, feelings and suggested changes with respect to the use of the tool and methodology.

Phase 2 of the emotion reporting tool was used again at a later date by an observer. During this phase of the study, the observer watched each piece of footage and independently gave their perception of the emotional change experienced by each of the participants at 30-second intervals, again systematically over each emotion.

Lastly, the video data was coded by a researcher along four dimensions. These were: changes in facial expression (frowning, smiling, breaking eye contact, as well as touching the face or hair), changes in body language / posture (sitting up, slouching, fidgeting), changes in vocalisations (intonation, speed of speech, fillers, long pauses) and lastly noting what was occurring within the context (such as, what the task was, who the participant was working with and in front of).

5. Results

The data presented is based on the 22 transcribed interviews. We provide a general description of the results space and in addition a detailed case study of one participant's complete data set. This analysis begins to reveal the reliability of the self-reports collected, as well as some of the inherent challenges and limitations of this approach.

5.1. How Do We Get under the Skin of Learners?

Three findings emerged from the 22 transcribed interviews. First, a number of the emotional terms used within the tool were inappropriate for use within the particular study context. For example, from a total of 22 participants, 12 reported that they never experienced anger at school and 6 commented that they never felt hopeless during the school day. In addition, the term "motivation" caused some confusion for the year 9 students, 6 of whom asked to have it explained to them.

Second, the interview data suggests, when given the choice, that the majority (well over half of those asked) would prefer to discuss and reflect on their emotional experience of learning with a group of friends, rather than at home on their own, with their teacher, or an independent adult. However, three of these participants believed completing this activity with their friends might render their reports less reliable.

Lastly, phase two of the tool requires the participants to compare the way they remember feeling during a certain segment of the video with how they remember feeling at the outset of the learning experience. Although most of the students believed the video helped them remember fairly accurately how they were feeling during the learning experience, fewer felt confident they could accurately remember how they were feeling at the outset of the lesson (prior to the video data being shot). Furthermore, some students reported feeling quite restricted by using the terms *More X*, *Less X* and *The same* when discussing their emotional experience. Their preferred means of discussing their comparative emotional experiences varied from either preferring to report on experience using a Likert scale type instrument, or introducing more specific qualitative terms into the second phase of the tool, such as "A Little More X", "A Lot More X" and so on (suggested solely by Yr 9 participants).

5.2. The Emotional Experience of Foreign Language Learning: Data Trends

One aim of this study was to develop an understanding of how a learner's emotional experience changes over time. From the data collected with our tool, it is possible to infer that emotions such as anxiety and embarrassment are liable to change significantly throughout an academic experience ($\chi 2$ = 37.4, p =.000, $\chi 2$ = 34.9, p =.000 respectively). The emotions most subject to large swings of change (i.e from More X to Less X within a 30 second segment) are relief and anxiety. The data also suggests that emotions such as boredom, hopelessness and anger are least likely to change, therefore if a learner starts a session feeling bored, the learning context is rarely able to alter this assessment.

5.2.1. A Case Study: Clare, a 17 Year Old Female A Level Student

The self-report data collected using our tool with Clare has been analysed and compared with two additional perspectives: that of the observer and that of the

researcher's video coding. This enables us to triangulate data sources to estimate the reliability of looking beyond self-report data collection for emotional experiences and provides evaluative data about the reliability of the data collected with the tool.

The observer's recorded beliefs about Clare's emotion change over each segment of the video data is presented in the middle section of Figure 1 and 2 alongside Clare's self-report data. The video data, coded by the researcher using four separate dimensions: changes in facial expression, changes in body language / posture, changes in vocalisations and lastly contextual changes is presented in the top section of Figure 1 and 2. The bottom section of Figure 1 and 2 depicts the emotional state as reported by Clare at the outset of the experience, using phase 1 of the emotion reporting tool.

The key difference between these two experiences can be found in Clare's success in her interactions. In Experience 1, we see Clare interacting successfully, albeit with the odd mistake, in contrast during Experience 2 she is unable to answer the question posed by the teacher and must watch whilst another student correctly answers the question. This perceived success can be correlated with a marked difference in visible emotional reaction, which is perhaps one reason why the tool appears more reliable at capturing her emotional experience in Experience 2. In other words, where there is an overt emotional reaction, the tool and methodology is capable of showing reliability.

However, what is perhaps more interesting are the less expected reported experiences. One would expect that as a student interacts successfully, and without many visible signs of discomfort, their confidence and pride and possibly their motivation would increase and their anxiety decrease. However, Clare reported feeling less confident, less proud and less motivated after a 30 second period of successful interaction with the teacher. Furthermore, if we compare Clare's reported emotional experience of being unable to answer the teacher's questions, with what Clare reports when she makes a small mistake (comparison of the Experience 1 at 30 sec, and Experience 2 at 30 sec), we see the reported emotional experience is similar.

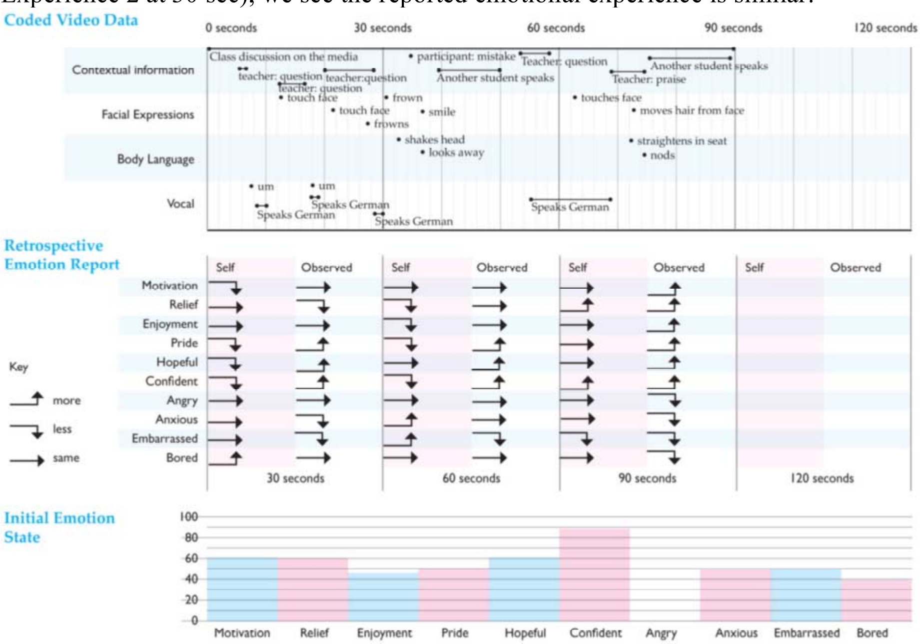


Figure 1. A graph comparing Clare's reported experience and the observed experience for Experience 1.

Figure 2. A graph comparing Clare's reported experience and the observed experience for Experience 2.

6. Discussion and Conclusion

This small study has shown there are clear and abundant discrepancies between what an observer believes the emotional reaction of a learner to be and what a learner reports their emotional reaction to be. In this case, who can we say is correct when it comes to interpreting a learner's emotional reaction and whose judgement should we rely on when adapting our educational interactions to the affective state of the learner? Clearly one contribution to the AIED community is to encourage the investigation of using multiple data sources when attempting to infer the emotional state of a learner.

This work also raises the question of whether there is a logical flow of emotions within a learning experience. If a student is interacting successfully with the teacher, displaying few visible signs of discomfort a logical model of emotional experience might conclude that this student, based on their context, is experiencing low anxiety and an increased feeling of pride and confidence, yet our study has found this logical flow of emotions is not what is experienced by our learner, or at least not what the learner reports they have experienced.

The findings of this study are furthermore capable of giving designers some insight into the differences in the temporal lives of emotions. For example, we have shown emotions such as boredom and hopelessness are less liable to rapid change over a learning session, and in contrast emotions such as relief and anxiety change rather more rapidly. This finding might be particularly interesting to designers of affectively intelligent systems if we correlate it with Craig's work [16] which concluded that boredom is negatively correlated with learning, since it provides the beginnings of a framework which starts to identify which learning emotions an educational system could *realistically* and with the most impact identify and react to.

When severe emotional changes are experienced, the tool, based on this case study, shows itself to be fairly reliable. However, it is entirely possible that when smaller emotional changes (as hypothesised to have occurred in Clare's emotional experience in Experience 1) are experienced, the tool becomes far less reliable. If affectively intelligent tools are to be effective they must be able to pick up on small changes in emotional experience which appear to be less prevalent in the type of body language, facial expression or vocalisations displayed by the learner, in order to avert some of the potentially larger changes in emotional experience.

A more detailed analysis of the self-report data alongside the video coding data together with triangulation will clarify the reliability of the data collection tool, in terms of the participant's self report of emotional experience versus an observer's report, as well as providing deeper insight into the extremities of emotional expression in the context of a formal learning setting.

References

[1] Del Soldato, T., du Boulay, B, *Implementation of motivational tactics in tutoring systems.* International Journal of Artificial Intelligence and Education, 1995. **6**: p. 337-378.

[2] Kort, B., Reilly, R. *Analytical Models of Emotions, Learning and Relationships: Towards an Affect-Sensitive Cognitive Machine*. in *Conference on Virtual Worlds and Simulation*. 2002. San Antonio, Texas.

[3] Porayska-Pomsta, K., Pain, H. *Providing cognitive and affective scaffolding through teaching strategies: applying linguistic politeness to the educational context*. in *Intelligent Tutoring Systems*. 2004. Brazil: Springer-Verlag.

[4] Zhang, L., Barnden, J., Hendley, R. *Affect detection and metaphor in E-Drama*. in *Workshop on motivation and affect in education software associated with the 12th international conference on artificial intelligence in education* 2004. Amsterdam: IOS press.

[5] Conati, C., Zhou, X. *Modelling students' emotions from cognitive appraisal in educational games*. in *ITS 2002*. 2002. Biarritz, France.

[6] Vygotsky, *Thought and language*. 1962, Cambridge, Mass.: The MIT Press.

[7] Vygotsky, *Mind in society: the development of higher psychological processes*, ed. M. Cole, Scribner, S., Souberman, E. 1978, Cambridge, Mass: Harvard University Press.

[8] Luckin, R., *Learning contexts as ecologies of resources: issues for the design of educational technology (CSRP Technical Report 578).* School of Cognitive Science Research Paper, 2005.

[9] Luckin, R., du Boulay, B., Smith, H., Underwood, J., Fitzpatrick, G., Holmberg, J., Kerawalla, L., Tunley, H., Brewster, D., Pearce, D, *Using mobile technology to create flexible learning contexts.* Journal of interactive media in education, 2005. **22**.

[10] Litman, D., Forbes-Riley, K. *Predicting student emotions in computer-human tutoring dialogues*. in *42nd annual meeting of the association for computational linguistics*. 2004. East Stroudsburg.

[11] Scherer, K., *What are emotions? And how can they be measured?* Social Science Information, 2005. **44**(4): p. 695-729.

[12] Do, S., Schallert, D, *Emotions and classroom talk: toward a model of the role of affect in students' experiences of classroom discussions.* Journal of educational psychology, 2004. **96**(4): p. 619-634.

[13] Pekrun, R., Goetz, T., Perry, R., Titz, W, *Academic emotions in students' self-regulated learning and achievement: a program of qualitative and quantitative research.* Educational psychologist, 2002. **37**(2): p. 91 - 105.

[14] O'Regan, K., *Emotion and e-learning.* JALN, 2003. **7**(3): p. 78-92.

[15] Wosnitza, M., Volet, S, *Origin, direction and impact of emotions in social online learning.* Learning and instruction, 2005. **15**: p. 449-464.

[16] Craig, S., Graesser, A., Sullins, J., Gholson, B, *Affect and learning: an exploratory look into the role of affect in learning with AutoTutor.* Journal of educational media, 2004. **29**(3): p. 241 - 250.

[17] Goleman, D., *Emotional intelligence: why it matters more than IQ*. 1996, London: Bloomsbury.

[18] Russell, J., *A circumplex model of affect.* Journal of personality and social psychology, 1980. **39**(6): p. 1161-1178.

[19] Cowie, R., Cornelius, R, *Describing the emotional states that are expressed in speech.* Speech Communication, 2003. **40**(1-2): p. 5-32.

Mind and Body: Dialogue and Posture for Affect Detection in Learning Environments

Sidney D'MELLO and Arthur GRAESSER
Institute for Intelligent Systems, The University of Memphis
365 Innovation Drive, University of Memphis, Memphis, TN, 38152, USA

Abstract. We investigated the potential of automatic detection of a learner's affective states from posture patterns and dialogue features obtained from an interaction with AutoTutor, an intelligent tutoring system with conversational dialogue. Training and validation data were collected from the sensors in a learning session with AutoTutor, after which the affective states of the learner were rated by the learner, a peer, and two trained judges. Machine learning experiments with several standard classifiers indicated that the dialogue and posture features could individually discriminate between the affective states of boredom, confusion, flow (engagement), and frustration. Our results also indicate that a combination of the dialogue and posture features does improve classification accuracy. However, the incremental gains associated with the combination of the two sensors were not sufficient to exhibit superadditivity (i.e., performance superior to an additive combination of individual channels). Instead, the combination of posture and dialogue reflected a modest amount of redundancy among these channels.

Keywords. Affect, emotions, tutorial dialogue, posture patterns, AutoTutor

1. Introduction

Learning inevitably involves emotions. Negative emotions (such as confusion, irritation, frustration, anger, and sometimes rage) are ordinarily associated with failure, making mistakes, diagnosing what went wrong, struggling with troublesome impasses, and starting a complex plan over again. Positive emotions (such as engagement, flow, delight, excitement and eureka) are experienced when tasks are completed, challenges are conquered, insights are unveiled, and major discoveries are made. We are convinced that a broad array of different emotions (or more generally, affect states) accompany the mastery of deep level concepts and complex learning. Consequently, we claim that Intelligent Tutoring Systems (ITS) should include a mechanism for motivating the learner, detecting a learner's emotional/motivational state, and appropriately responding to that state [1, 2, 3].

One of the challenges of such an endeavor involves developing computational systems to reliably detect the learner's emotions. An emotion classifier need not be perfect but should have some modicum of accuracy. Although emotion recognition is an extremely difficult problem, on par with automatic speech recognition, a number of sustained efforts have yielded encouraging results. The majority of these affect detection systems are based on monitoring physiological signals, such as galvanic skin response, heart rate, or other bodily measures that include facial expressions, voice stress analysis, and acoustic-prosodic features [see 4 for a comprehensive review]. As an interesting

alternative, this paper explores the possibility of affect detection from two relatively unexplored sensors: dialogue features and posture patterns. Features from these sensors were extracted from an interaction session with AutoTutor, an ITS that helps learners construct explanations by interacting with them in natural language [5]. The overall goal of the project is to transform AutoTutor into an affect-sensitive intelligent tutoring system.

There are good reasons for expecting that dialogue features are diagnostic of affect in learning environments. The dialogue in one-on-one tutoring sessions yields a rich trace of contextual information, characteristics of the learner, episodes during the coverage of the topic, and social dynamics between the tutor and learner. Within the context of tutoring systems, we would predict that dialogue features provide a very robust diagnostic channel to infer a learner's affect because it has both a broad and deep feature set that covers deep meaning, world knowledge, and pragmatic aspects of communication.

Our use of posture for affect detection is motivated by a number of embodied theories of cognition [e.g. 6, 7]. Theories of embodied cognition postulate that cognitive processes are constrained substantially by the environment and by the coupling of perception and action. If embodied theories are correct, the cognitive and emotional states of a learner are implicitly or intentionally manifested through their gross body language. Therefore, the monitoring of posture patterns may lead to insights about the corresponding cognitive states and affective arousal. An added advantage of monitoring posture patterns is that these motions are ordinarily unconscious and thereby not susceptible to social editing, at least compared with speech acts in conversation.

The fact that two sensors will be monitored raises the issue of how the sensory fusion will impact emotion classification accuracy. One intriguing hypothesis is that classification performance from multiple channels will exhibit super-additivity, that is, classification performance from multiple channels will be superior to an additive combination of individual channels. Simply put, the whole will be greater than the sum of the parts. An alternative hypothesis would be that there is redundancy between the channels. When there is redundancy, the addition of one channel to another channel yields negligible incremental gains; the features of the two channels are manifestations of very similar mechanisms.

Much of the known research in affect detection has involved the use of a single modality to infer affect. Some notable exceptions involve the use of four physiological signals to detect 8 basic emotions [8] and various combinations of audio-visual features [9, 10, 11]. More recently, and of relevance to the context of learning, Kapoor and Picard [12] developed a probabilistic system to infer a child's interest level on the basis of upper and lower facial feature tracking, posture patterns (current posture and level of activity), and some contextual information (difficulty level and state of the game). The combination of these modalities yielded a recognition accuracy of 86%, which was quantitatively greater than that achieved from the facial features (67% upper face, 53% lower face) and contextual information (57%). However, the posture features alone yielded an accuracy of 82% which would indicate that posture was redundant with the other channels.

Before proceeding with a description of the experiments on automated affect detection, it is important to identify some differences between our approach and previously established research in this area. Compared to previous research on affect detection, we monitored a larger number of learning-relevant affective states (N = 7) and attempted to automatically distinguish between a subset (N=4) of these. We collected affect judgments

from multiple human judges in order to establish ground truth (or a gold standard) regarding the learners' affect. This can be contrasted with a number of researchers who have relied on a single operational measure when inferring a learner's emotion, such as self reports or ratings by independent judges. The present data collection procedure also used ecologically valid methods while monitoring the affect of a learner, namely learners interacting with AutoTutor in a bona fide tutoring session. No actors were used and no attempts were made to intentionally invoke affect. Finally, this research is novel by virtue of tracking relatively unexplored sensors to detect affect, namely dialogue and posture.

2. Empirical Data Collection

Our study collected tutorial dialogues and posture patterns from 28 college students who interacted with AutoTutor for 32 minutes on one of three randomly assigned topics in computer literacy: hardware, internet, or operating systems. During the interaction process, a video of the participant's face and a video of the screen were recorded. The gross body language of the learner was also recorded using the Body Pressure Measurement System (BPMS, described below). The judging process was initiated by synchronizing the video streams from the screen and the face and displaying them to the judge. Judges were instructed to make judgments on what affective states were present in 20-second intervals, at which time the video automatically paused (freeze-framed). They were also instructed to indicate any affective states that were present in between the 20-second stops.

Four sets of emotion judgments were made for the observed affective states of each participant's AutoTutor session. For the self judgments, the participant watched his or her own session with AutoTutor immediately after having interacted with the tutor. For the peer judgments, the participants returned approximately a week later to watch and judge another participant's session on the same topic in computer literacy. Finally, two additional judges (called trained judges), who had been trained on how to detect facial action units according to Paul Ekman's Facial Action Coding System (FACS) [13], judged all of the sessions separately. The trained judges also had considerable interaction experience with AutoTutor. Hence, their emotion judgments were based on contextual dialogue information as well as the FACS system

A list of the affective states and definitions was provided for all judges. The states were boredom, confusion, flow, frustration, delight, neutral and surprise. The selection of emotions was based on previous studies of AutoTutor [14, 15] that collected observational data (i.e., trained judges observing learners) and emote aloud protocols while college students learned with AutoTutor.

Interjudge reliability was computed using Cohen's kappa for all possible pairs of judges: self, peer, trained judge1, and trained judge2. Cohen's kappa measures the proportion of agreements between two judges, with correction for baserate levels and random guessing. There were 6 possible pairs altogether. The kappa's were reported in Graesser et al. [16]: self-peer (.08), self-judge1 (.14), self-judge2 (.16), peer-judge1 (.14), peer-judge2 (.18), and judge1-judge2 (.36). These kappa scores revealed that the trained judges had the highest agreement, the self-peer pair had lowest agreement, and the other pairs of judges were in between. It should be noted, however, that the kappa scores increase substantially (.12, .31, .24, .36, .37, and .71) when we focus on observations

in which the learner declares they have an emotion, as opposed to points when they are essentially neutral. The kappa scores are on par with data reported by other researchers who have assessed identification of emotions by humans [e.g. 17].

3. Automated Affect Detection from Dialogue and Posture

One important question that arises during the design of a multisensory emotion classification system involves determining the appropriate level of abstraction at which to fuse the output from the sensors. Fusion at the feature level involves grouping features from the various sensors before attempting to classify emotions. Alternatively, in decision level fusion, the affective states would first be classified from each sensor and then integrated to obtain a global view across the various sensors. While decision level fusion is more common in HCI applications [18], Pantic and Rothkrantz [4] have questioned the validity of using decision level fusion in the affective domain because audio, visual, and tactile signals of a user are typically displayed in conjunction and with some redundancy. Consequently, in this paper we explore the use of feature level fusion alone. We also restrict our scope to a naïve fusion algorithm in which the features from the individual sensors are additively combined before attempting classification.

3.1. Dialogue Features

We mined several features from AutoTutor's log files in order to explore the links between the dialogue features and the affective states of the learners (see [15], for additional details about mining the dialogues). These features included temporal assessments for each student-tutor turn, such as the subtopic number, the turn number, and the student's reaction time (interval between presentation of the question and the submission of the student's answer). Assessments of response verbosity included the number of characters (letters, numbers) and speech act (that is, whether the student's speech act was a contribution towards an answer which was coded as a 1 versus a frozen expression, e.g., "I don't know, " "Uh huh," coded as -1). The conceptual quality of the student's response was represented by 4 features obtained with Latent Semantic Analysis (LSA, [19]). LSA is a statistical technique that measures the conceptual similarity of two text sources. AutoTutor's major dialogue moves were ordered onto a scale of conversational directness, ranging from -1 to 1, in terms of the amount of information the tutor explicitly provides the student (e.g. pumps < hints < prompts < assertions < summaries). AutoTutor's short feedback (negative, neutral, and positive) was displayed in its verbal content, intonation, and a host of other non-verbal cues. The feedback was transformed to a 5-point scale ranging from -1 (negative feedback) to 1 (positive feedback).

3.2. Posture Features

The Body Posture Measurement System (BPMS), developed by Tekscan™, was used to monitor the gross body language of a student during a session with AutoTutor. The BPMS consists of a thin-film pressure pad (or mat) that can be mounted on a variety of surfaces. The output of the BPMS system consisted of two 38x41 matrices (for the back and seat) with each cell in the matrix corresponding to the amount of pressure exerted on

the corresponding element in the sensor grid.

Several features were computed by analyzing the pressure maps of the 28 participants recorded in the study. We computed 5 pressure related features and 2 features related to the pressure coverage for both the back and the seat, yielding 14 features in all. Each of the features was computed by examining the pressure map during an emotional episode (called the current frame). The pressure-related features include the net pressure, which measures the average pressure exerted. The prior change and post change measure the difference between the net pressure in the current frame and the frame three seconds earlier or later, respectively. The reference change measures the difference between the net pressure in the current frame and the frame for the last known affective rating. Finally, the net pressure change measures the mean change in the net pressure across a predefined window, typically 4 seconds, that spans two seconds before and two seconds after an emotion judgment. The two coverage features examined the proportion of non-negative sensing units (net coverage) on each pad along with the mean change of this feature across a 4-second window (net coverage change).

3.3. Classification Experiments

Four data sets, corresponding to each of the four judge's emotion judgments, were obtained by extracting the set of dialogue features for each turn and temporally integrating them with the emotion judgments. More specifically, the emotion judgment that immediately followed a dialogue move (within a 15 second interval) was bound to that dialogue move. Similarly, posture features were extracted from the pressure maps obtained at the time of the emotional experience. This allowed us to obtain four sets of labeled dialogue and posture data aggregated across the 28 participants. The sizes of these data sets were 1022, 1038, 1113, and 1107 for affect labels provided by the self, the peer, trained judge1, and trained judge2, respectively.

The Waikato Environment for Knowledge Analysis (WEKA) was used to evaluate the performance of various standard classification techniques in an attempt to detect affect from dialogue and posture. The classifiers used were a Naïve Bayes classifier, a simple logistic regression, support vector machines, k-nearest neighbor with k = 1, C4.5 decision trees, and an additive logistic regression meta classifier. The classification algorithms were compared in their ability to discriminate between boredom, confusion, flow, and frustration. Delight and surprise was excluded due to a comparatively low number of observations. To establish a uniform baseline, we randomly sampled an equal number of observations from each affective state category. This process was repeated for 10 iterations and all reported reliability statistics were averaged across these 10 iterations. Classification reliability was evaluated on the 6 classification algorithms using k-fold cross-validation (k = 10).

Figure 1 presents kappa scores (classification accuracy after adjusting for the 25% base rate), averaged across the 6 classifiers. The discrimination was among the affective states of boredom, confusion, flow, and frustration. These results support a number of conclusions regarding automated affect detection. First, both dialogue and posture were moderately successful in discriminating between the affective states. One may object to our use of the term *moderate* to characterize our classification results. However, it is imperative to note that an upper bound on automated classification accuracy of affect has yet to be established. While human classifications may be considered to be the ultimate

upper bound on system performance, human performance is variable and not necessarily the best gold standard. In particular, the interrater reliability (kappa) scores between the various human judges for a subset of the affective states (boredom, confusion, flow, and frustration) was: kappa$_{\text{self-peer}}$ = .13, kappa$_{\text{self-judge1}}$ = .30, kappa$_{\text{self-judge2}}$ = .29, kappa$_{\text{peer-judge1}}$ = .33, kappa$_{\text{peer-judge2}}$ = .34, and kappa$_{\text{judge1-judge2}}$ = .58. The highest kappa score for the machine generated emotion categories was kappa$_{\text{machine}}$ = .32 (or 50% accuracy for a four-way classification). This places the affect detection capabilities of the computer on par with novice judges (self and peer), but significantly lower than trained judges.

Statisticians have sometimes claimed, with hedges and qualifications, that kappa scores ranging from 0.4 – 0.6 are typically considered to be fair, 0.6 – 0.75 are good, and scores greater than 0.75 are excellent [20]. On the basis of this categorization, the kappa scores obtained by our best classifier (.32) would range from poor to fair. However, such claims of statisticians address the reliability of multiple judges or sensors when the researcher is asserting that the decisions are clear-cut and decidable. The present goal is very different. Instead, our goal is to use the kappa score as an unbiased metric of the reliability of making affect decisions, knowing full well that such judgments are fuzzy, ill-defined, and possibly indeterminate. A kappa score greater than 0.6 is expected when judges code some simple human behaviors, such as facial action units, basic gestures, and other visible behavior. However, in our case the human judges and computer algorithms are inferring a complex mental state. Moreover, it is the relative magnitude of these measures among judges, sensors, and conditions that matter, not the absolute magnitude of the scores. We argue that the lower kappa scores are meaningful and interpretable as dependent measures (as opposed to checks for reliability of coding), especially since it is unlikely that perfect agreement will ever be achieved and there is no objective gold standard..

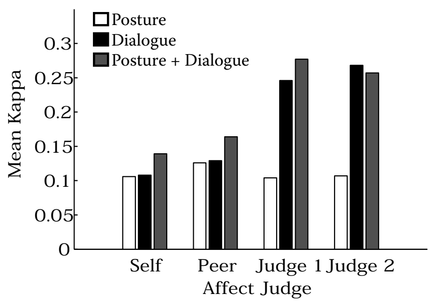


Figure 1. Classification Accuracy

The results unveil interesting patterns between the various sensors and the type of judge that provided the emotion ratings. We find that the use of posture features yielded an average kappa of .112 for the novice datasets (self and peer), which is on par with the kappa obtained on the data sets in which affect judgments were provided by the trained judges (.106). For the dialogue features, we note that kappa scores associated with classifiers trained on the novice data sets (.119) were quantitatively lower than the kappa obtained from the trained judges' data sets (.257). One possible explanation for this difference is that the trained judges had considerable experience interacting with AutoTutor and conceivably tapped into this knowledge while they rated the learner's emotions. This use of contextual knowledge coupled with a video of the participant's face is reflected in the kappa scores obtained by the classifiers.

The results also indicate that a simple combination of dialogue and posture features does yield performance gains. That is, the combination of two sensors is higher than either one alone. However, it is unclear as to whether these performance gains reflect a degree of superadditivity or redundancy. One possible metric to assess superadditivity is on the basis of assessing improvements afforded by multisensory fusion over and above the maximum unisensory response. Specifically, if k_p and k_d are the kappa scores

associated with detecting affect when posture and dialogue are considered individually, and k_{p+d} is the kappa score associated with a combination of these two channels, then the degree of superadditivity obtained would be $[(k_{p+d} - \max(k_{p,}\ k_{d})]/\max(k_{p,}\ k_{d})$. This metric is widely used by neuroscientists studying multisensory integration with respect to visual, audio, and tactile senses in humans and animals [21]. The use of this metric to assess superadditivity yielded incremental gains of .3, .3, .1, and 0 from the data sets of the self, peer, and the 2 trained judges.

As an alternative, one could consider incremental gains obtained above and beyond an additive combination of the two sensors. This can be expressed as $k_{add} = k_{p} + k_{d} - k_{p} * k_{d}$. Superadditivity would be achieved if $k_{p+d} > k_{add}$, additivity if $k_{p+d} = k_{add.}$ Redundancy occurs in situations where the combination of two or more sensory channels does not produce incremental gains (i.e. $k_{p+d} < k_{add}$). On the basis of this more conservative metric we conclude that a combination of posture and dialogue features resulted in redundancy.

4. Discussion

Multimodal systems for affect detection in general user modeling has been widely advocated but rarely implemented [22]. This is mainly due to the inherent challenges with unisensory affect detection, which no doubt increase in multisensor environments. The present research is a modest but important step in affect detection research.

This paper investigated an affect classifier to discriminate among the affective states of boredom, confusion, flow, and frustration. Although, this classifier did not consider neutral affect, neutral observations will be important to consider in the next step of the project. The next step will adaptively tailor AutoTutor's dialogue moves to the learner's affective states in addition to their cognitive states. In our view, the comparative classifier developed here would serve as a tie-breaker for several separate individual affect-neutral classifiers. Specifically, we envision a set of affect-neutral classifiers that determine whether the learner is experiencing an emotion (boredom, confusion, flow, or frustration) or is in the neutral state. If there is resonance with one and only one emotion, then that emotion is declared as being experienced by the learner. If there is resonance with 2 or more emotions, then the comparative classifier developed in this study would be used to discriminate among candidate emotions. We are currently in the process of calibrating the reliability of such affect-neutral classifiers. Our most recent results reveal that kappas associated with detecting each emotion from neutral are fair, at least with respect to the criterion established above: boredom = .40, confusion = .52, flow = .48, and frustration = .54. More details of this analysis are reported in D'Mello, Picard, and Graesser [23].

The broader scope of this research ventures into the areas of embodied cognition. The discovery of redundancy between the dialogue and posture features indicate that there are significant relationships between cognition and bodily movements. However, many research questions remain unanswered. To what extent can affect and cognition individually predict bodily activity? Does a combination of these channels increase their predictive power?. Answers to these questions will help us explore theories of embodied cognition in addition to the synchronization of emotions with complex learning.

Acknowledgements

This research was supported by the National Science Foundation (REC 0106965 , ITR

0325428, and REC 0633918). Any opinions, findings and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of NSF.

References

[1] Picard, R. (1997). Affective Computing. The MIT Press, Cambridge, MA.

[2] Issroff, K. & del Soldato, T. (1996). Incorporating motivation into computer-supported collaborative learning. *Proceedings of the European Conference on Artificial Intelligence in Education.*

[3] Lepper. M, R., & Chabay, R.W. (1988). Socializing the intelligent tutor: Bringing empathy to computer tutors. In H. Mandl & A. Lesgold (Eds.), *Learning Issues for Intelligent Tutoring Systems* (pp. 242-257). Hillsdale, NJ: Erlbaum.

[4] Pantic, M., & Rothkrantz, L. J. M. (2003). Towards an affect-sensitive multimodal human-computer interaction. *IEEE Special Issue on Multimodal Human-Computer Interaction, 91*(9), 370-1390.

[5] Graesser, A., Wiemer-Hastings, K., Wiemer-Hastings, P., Kreuz, R., & the Tutoring Research Group (2000). AutoTutor: A simulation of a human tutor. *Journal of Cognitive Systems Research, 1,* 35-51.

[6] Clark, A. (1997). *Being there: Putting brain body and world together again*. Cambridge, MIT Press.

[7] Glenberg, A. M., & Kaschak, M. (2002). Grounding language in action. *Psychonomic Bulletin & Review*, *9*, 558-565.

[8] Picard, R. W., Vyzas, E., & Healey, J. (2001). Toward machine emotional intelligence: Analysis of affective physiological state. *IEEE Transactions Pattern Analysis and Machine Intelligence, 23*(10), 1175-1191

[9] Chen, L. S., Huang, T. S., Miyasato, T., & Nakatsu, R. (1998). Multimodal human emotion/expression recognition. In *Proc. FG* (pp. 396–401).

[10] Dasarathy, B. V. (1997). Sensor fusion potential exploitation: Innovative architectures and illustrative approaches. *Proc. IEEE, 85*, 24-38.

[11] Yoshitomi, Y., Kim, S., Kawano, T., & Kitazoe, T. (2000). Effect of sensor fusion for recognition of emotional states using voice, face image and thermal image of face. In *Proc. ROMAN* (pp. 178–183).

[12] Kapoor, A. and Picard, R. E. (2005), Multimodal Affect Recognition in Learning Environments, ACM MM'05, November 6-11, 2005, Singapore.

[13] Ekman, P, & Friesen, W. V. (1978). *The facial action coding system: A technique for the measurement of facial movement*. Palo Alto: Consulting Psychologists Press.

[14] Craig, S. D., Graesser, A. C., Sullins, J., & Gholson, B. (2004). Affect and learning: An exploratory look into the role of affect in learning. *Journal of Educational Media, 29*, 241-250.

[15] D'Mello, S. K., Craig, S. D., Sullins, J., & Graesser, A. C. (2006). Predicting affective states through an emote-aloud procedure from AutoTutor's mixed-initiative dialogue. *International Journal of Artificial Intelligence in Education, 16*, 3-28.

[16] Graesser, A.C., McDaniel, B., Chipman, P., Witherspoon, A., D'Mello, S., & Gholson, B. (2006). Detection of emotions during learning with AutoTutor. In R. Son (Ed.), *Proceedings of the 28th Annual Meetings of the Cognitive Science Society* (pp. 285-290). Mahwah, NJ: Erlbaum.

[17] Marsic, I., Medl, A., and Flanagan, J. L. (2000) Natural Communication with Information Systems, *Proceedings of the IEEE*, 88(8), 1354-1366.

[18] Pentland, A. (2000). Looking at people: sensing for ubiquitous and sensable computing. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22, 107-119.

[19] Landauer, T. K., & Dumais, S. T. (1997). A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review,104*, 211-240.

[20] Robson C. (1993). *Real word research: A resource for social scientist and practitioner researchers*. Oxford: Blackwell.

[21] Holmes, N., & Spence, C. (2005). Multisensory Integration: Space, Time and Superadditivity. *Current Biology, 15*(18), 762-764.

[22] Jaimes, A., & Sebe, N. (2005) Multimodal human computer interaction: A survey. *IEEE International Workshop on Human-Computer Interaction (HCI 2005),* Beijing, China.

[23] D'Mello, S. K., Picard, R., & Graesser, A. C. (in review). Towards an affect-sensitive AutoTutor.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Towards Predictive Modelling of Student Affect from Web-Based Interactions

Manolis MAVRIKIS [a,b], Antony MACIOCIA [a] and John LEE [b]
[a] *School of Mathematics, The University of Edinburgh* [1]
[b] *School of Informatics, The University of Edinburgh*

Abstract.
This paper presents the methodology and results of a study conducted in order to establish ways of predicting students' emotional and motivational states while they are working with Interactive Learning Environments (ILEs). The interactions of a group of students using, under realistic circumstances, an ILE were recorded and replayed to them during post-task walkthroughs. With the help of machine learning we determine patterns that contribute to the overall task of diagnosing learners' affective states based on observable student-system interactions. Apart from the specific rules brought forward, we present our work as a general method of deriving predictive rules or, when there is not enough evidence, generate at least hypotheses that can guide further research.
Keywords. Affective modelling, machine learning

1. Introduction

The growing interest in affective computing in general [24] has influenced AIED in particular. Researchers are interested in engaging educational systems and learners in a more naturalistic interaction through an affective loop [10] that involves the *detection* of a learner's affective state, *selection* of appropriate actions and *synthesis* of emotional expressions [10,12]. Our interest lies primarily in enhancing ILEs with the ability to detect and adapt to emotional and motivational states of the learner. However, we believe that researchers do not know enough about the exact ways students interact with ILEs, particularly when working in their own time rather than when participating in a study where the social dynamics can be different. This leads to systems that are not necessarily tuned to students' actual behaviours. In our research, we endeavour to drive our design guided by empirical studies whilst acknowledging the inherent difficulty and complexity of conducting such studies especially when investigating affective aspects.

As also highlighted in [9] different students in the same situations have different affective reactions which are not always caused directly by the material or the behaviour of the (system or human) tutor. Factors such as goals, preferences, personality [9], students' attitude towards computers [6], their own view of their knowledge state [23] and, as we have observed in previous pilots, the context of the study itself play an important role. Ignoring these factors undermines the ecological validity of any study, yielding results which are not transferable to the situations under which the educational systems are used. Therefore, in our research we investigate a combination of affective and moti-

[1] Correspondence to: JCMB, Mayfield Road, Edinburgh EH9 3JZ, UK. Email: m.mavrikis@ed.ac.uk

vational states, which accompany processes involving learning, whilst trying to conduct as ecologically valid studies as possible by integrating them in realistic situations rather than artificial ones that could change students' emotions and motivation.

This paper presents such a study integrated into a course during which students use an ILE as additional support. The main goal of the study, and of the subsequent analysis, is to investigate the possibility of predicting students' emotional and motivational states based on student-system interactions and therefore, to contribute to the detection part of the affective loop. Student actions are the least obtrusive input stream possible (compared to facial expressions, voice or other bodily measures [16,25]) and constitute the way students interact with the system anyway so they are inexpensive to collect. In addition, investigating the actual way students interact with the ILE is the first step towards implementing any effective pedagogical agent that takes decisions and adapts its behaviour. A means of reaching this understanding is to enrich the results of traditional knowledge elicitation approaches by employing machine learning techniques [4,5,20]. In the current approach we employ decision trees that can help the underlying relationship of the data emerge and, in contrast to other techniques, provide more inspectable results that can be easily transformed into rules and hence can also be implemented straightforwardly.

2. The ILE used in the study

WaLLiS [19], is a web-based ILE used by the School of Mathematics of The University of Edinburgh as a way of addressing the difficulties school students face when entering university. The system hosts content such as theory or examples as well as interactive exercises. The exercises comprise of activities in the form of applets during which students interact with a microworld [22] exploring a concept (e.g a dual cone to understand the different conic sections). In addition, there are other exercises which are more constrained, cloze or multiple choice questions with multiple steps which are delivered as a web page with interactive elements. A tree map of the contents gives students control over which part of the material to study. The important feature of the system for our research is the feedback mechanism which relies on a Computer Algebra System for the interactive exercises and a geometric engine for the exploratory ones. The mechanism is mainly inspired by theories of cognitive skill acquisition (such as ACT-R [3]) and cognitive scaffolding [31]. Similar to the CMU Cognitive Tutors [2] help is offered in an instrumental way taking into account students' misconceptions and providing as much help as necessary for them to progress. This way a problem which students cannot solve is turned into a teaching aid from which they learn by practice. Their answers are evaluated and appropriate feedback is given. When the answer is correct they can move on to the next step or request the solution. Otherwise, they are provided with corrective feedback (if their misconception is known) or negative feedback with a suggestion to try again. An exercise step is usually associated with 3 to 5 hints . The first few rephrase the question statement and give instructions on how to proceed or resolve their misconception while the last is almost providing the answer. At the end, students can request the solution. A similar approach is followed during the exploratory activities.

In addition, there is another level of help that deals with the global history of students' interaction with the system and suggests what page they should study. Finally, during an exercise students can still access theory and example. This provides a decontextual (or as referred in other systems 'glossary' [1]) help. The pages are either proposed by the the suggestion mechanism (if the system thinks that the student did not spend the necessary time in the relevant page) or can be accessed by students' initiative on demand.

3. The study

3.1. The selection of the participants

Participants were selected from a group of approximately 200 undergraduates who attend the "Geometry and Convergence" course. Time and budget constrains limited the number of participants to 20. Therefore, we wanted to select an appropriate representative sample. Groups of students were created based on entry qualification and their marks in a prerequisite course. Similar to performing a disproportionate stratified sampling [17] volunteers were taken from each group. A further selection took place using a pre-test of 10 questions targeting knowledge or familiarity of conics sections. All students failed the pre-test and only 3 claimed to be familiar with the concept of conics in advance. From the selected students only 18 (3 with very low previous ability, 5 low, 5 medium, 3 high and 2 very high) completed all necessary tasks and faced no technical problems.

3.2. Data gathering for building predictive models

To collect as realistic data as possible we chose a particular part of the 'Geometry and Convergence' course that deals with Conic Sections and has been taught successfully in the past using WALLIS rather than the conventional lectures. For the purposes of the study, we allowed students to work in their natural setting (i.e. their own time and location). This decision introduces technical and methodological problems. For example, it is difficult to record all aspects of student-system interaction, or employ methods like 'talk-aloud' protocols. To avoid these problems a functionality was added to the system to unobtrusively record students' interactions such as button and link clicks, and even mouse movements [18]. These are stored in a remote server and can be replayed later. In addition, students were given a short questionnaire in advance with screen shots of the system's pages and space for note-taking, which they were asked to complete after each interaction with the system to interfere as little as possible with the learning process. This methodology resembles 'confidence logs' or similar techniques of note-taking that increase metacognitive skills in educational studies [13]. The notes helped stimulate students' memory and served as a guide for the next stage, a post-task walkthrough, similar to a cognitive walkthrough but particularly aiming to record changes in students' affect.

3.3. Situational factors and their values

We focus our investigation on a set of factors encapsulated under the title *situational factors* [26]. Their importance and possible values were inspired from related literature [5,9,12,15,27] and our previous studies and pilots. First of all we use 3 *emotional states* (frustration, boredom, confusion) which take binary values. We identified that students find it easier to report the presence or absence of an emotion rather than its intensity. Moreover, a general category for *positive feelings* represents satisfaction, joy, happiness etc. In previous pilots this helped students, who often found it difficult to verbalise or differentiate their feelings, to be more expressive. In addition, we employ 3 relative change factors: *confidence*, *interest* and *effort*. Students found it easier to talk in terms of changes rather than choosing a specific value from a range. Since our interest lies in detecting and dealing with boundary situations, these are satisfactory to record. This, also facilitates the machine learning analysis as we acquire more instances of the same situation. Finally, we include factors that relate to and depend on the situation: *difficulty* of the current material or step, *time spent* to complete a task measured in standard deviations from the corresponding mean, *correctness* of an answer, *interaction type* (input box, radio button etc.) and *student characteristics* (e.g. previous ability).

3.4. Process

Students were asked to comment on the situational factors during semi-structured replays. If it seemed that they were forgetting to express themselves they were reminded of this. For every state expressed a timestamp was associated with the log file. Students were strongly encouraged to express anything in relation to affect or motivation even if it was not included in the description of the situational factors that was given to them. In places which the students had recorded in the log sheet, or which we considered important, the replay was paused to allow us to elaborate on particular issues and gain additional insight in order to guide the subsequent data analysis. Apart from the fact that we wanted to confirm that there are no occurrences caused by characteristics of the system (e.g. frustration from delays or confusions because of the interface), the semi-structure of the 'interview' meant that we could disambiguate the source behind the reported factor and particularly the action or goal of the student at that moment.

4. Analysis and Results

The raw data were converted to spreadsheets by an extraction tool that matched the timestamps of the reported situational factors with factor values derived from the system (e.g. correctness). As a first step towards specifying patterns and trends of the data we performed frequency analysis as well as correlations of the expressed factor value against the immediately preceding student-system action. This part was inspired by similar work described in [12] where the affective states as reported by students during talk-aloud protocols were correlated with the system actions. Because the results of both our descriptive analysis and the general directions of correlations are very similar we point the interested reader to [12]. Here we focus our attempts on improving the accuracy of the predictions by taking into account *historical* information to determine what drives students to report situational factor changes. For example, *hint detail* is significantly correlated (r=0.63) with expressions of increased *confidence* (i.e confidence increases with detailed hints and full solutions rather than simple hints) and negatively correlated (r=-0.59) with *confusion* (i.e less detailed hints confuse students more). As explained before, feedback occurs after students' request for hint. Guided by the walkthroughs we removed the instances that occur after students ask for hints too quickly. This increases the correlations to $r = 0.72$ and $r = -0.64$ respectively. The main reason, as we found out during the replays, is that in this more naturalistic setting, some students abuse the hint facility in order to get to the solution. A similar pattern of behaviour is reported in [5] and referred to as 'gaming the system'. The fact that students never read these hints does not help their confidence or confusion. This aspect would have never been captured if looking only at the immediately preceding student action. Other examples include asking for hint after having tried to answer many times rather than asking for hints immediately. A similar problem is reported in [12] in relation to boredom detection and the absence of historical data. It seems intuitively necessary, but examples such as the above and interviews with tutors and students in other studies (e.g. [27]) make more evident that the history of actions plays an important role in improving the predictions of the situational factors.

4.1. Predicting factors from interactions

In order to include history in our predictive model and for the reasons explained in the Introduction we employ decision trees. We use the tree induction algorithm J4.8 of WEKA

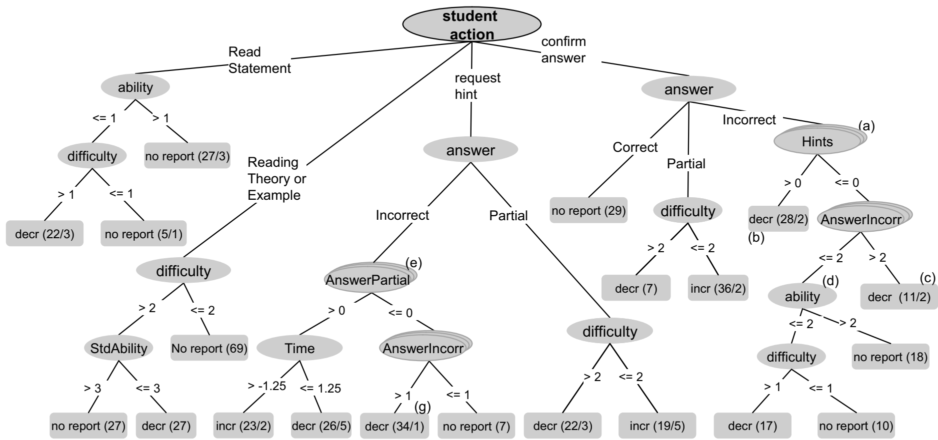


Figure 1. Graphical representation of part of the decision tree for confidence. Following a path from root to a leaf creates a rule that shows the value of the factor given the values of the attributes in the path. The numbers in brackets show the amount of instances correctly classified by this rule over the misclassified ones. Nodes with layers (eg. node a) originate from the vector of historical attributes. For example `AnswerIncorr > 2` shows that in the relevant time window more than two incorrect answers were provided by the student.

[30], which is based on improvements of Quinlan's popular algorithm; C4.5. For each situational factor of our interest the algorithm is presented with pre-processed vectors (instances) derived from the raw data. These vectors consist of features (such as *difficulty* or last *action*) and a nominal class that encodes the factor we want to predict. For *confidence*, *interest*, and *effort* the class takes values that depict relative or extreme changes: *decrease, increase* and *extr_decrease, extr_increase*. For the *emotional state* factor the values are the different emotions described in section 3.3. Depending on the prediction task, different features are chosen based on our descriptive and correlation analysis. The history is represented as a vector the elements of which encode the number of times that each type of possible action (eg. hint) occurred in a relevant time window. This spans back to the last change of the factor under investigation or (if it has not recently changed) to the start of the relevant situation or exercise. Due to space limitations we use here *confidence* as a case study to depict the methodology and the type of results we obtain.

Predicting students' confidence

In total there were 289 reports in relation to confidence; these comprise set A. Running the algorithm on this set of data results in biased rules that are influenced by the fact that only changes are reported. Therefore, we extract the instances (249) where the same actions as these of set A occur but are not associated with a particular report. Figure 1 shows a graphic representation of the tree resulting from the merged sets of instances with attributes *action*, *difficulty*, student *ability*, last *answer*, *time* between the last two action, and the *history* vector. 90.9% of the cases are classified correctly and Cohen's Kappa [8] is 0.87 which can be considered a satisfactory result.

As an example, we describe the rules associated with confirm answer actions. The data suggest that when students confirm an incorrect answer (node a) having previously requested at least one hint, their confidence decreases (leaf b). If no hints were requested and if they previously had many (more than 2) incorrect answers then they report that their confidence decreases (the two misclassified instances in leaf c are of an extreme

decrease). Otherwise it seems that the outcome depends on students' ability and the difficulty of the question (node d). Similarly we can interpret the rest of the tree. Notice the rule associated with node (e) where students are requesting hints after submitting a wrong answer. Their reaction depends on having previously replied partially correct. If they have done so, the fact that they now receive negative feedback seems to dent the confidence of the students who have not spent enough time to read the hint (time is in standard deviations below the mean). The others probably get over this problem by spending more time in trying to understand. Note that the same pattern is associated with increased effort in the effort tree. The remaining rules seem, in general, equally intuitively plausible. However, in some cases (especially where misclassifications occur) detailed analysis of the raw data has to be conducted in order to establish the reasons behing them. This process generates hypotheses for further studies.

We conclude our example with a common problem. The rule ending with leaf g indicates that, when requesting a hint for an incorrect answer and if the learner had replied incorrectly more than once in the past but no partially correct answers were given then their confidence decreases. However, there are no reports for the cases where there was only one (or none) incorrect answers before. This raises a question. In most of the cases where this happens the reason is that either the student commented on another factor (e.g.the *emotional state* factor or their *interest*) or that simply there was no change to the factor. However, as we further explain in the Discussion section, there are also cases where the absence of reports was due to the lack of metacognitive skills of the student. As an aside, it is worth noting here that in 63% of the cases where the behaviour captured by the above rule was observed, students attempted to quit the page. This is another example that demonstrates the importance of adapting the environment according to correctly diagnosed students' affective states. In this case the system could intervene appropriately and attempt to boost student's confidence instead of letting them quit.

5. Discussion

The methodology described in this paper and the study conducted yielded useful results that contribute to the overall task of detecting learners' emotional and motivational state. First of all, by recording as realistic interactions as possible we learn more about the way students interact with the system. Through walkthroughs and interviews we identify patterns of student-system interactions where students' emotions and motivation are mostly affected. Rule induction provides a mechanism for deriving a predictive model in the form of rules that are based on students' actions. These rules seem intuitive plausible. However, it would be difficult to generate all these rules intuitively and the fact that they are derived from data makes them more trustworthy. For every factor of interest we derive an inspectable tree which can form part of a predictive model and could be included in other frameworks (e.g. [9,12,21]) contributing to the overall diagnosis of the learner's state. To improve its accuracy a microanalysis of misclassified rules is needed. Despite the fact that it is time consuming it can help improve the quality of the results in relation to the diagnostics process of a system and consequently make it more effective. Moreover, even when there are only a few instances to justify the existence of a rule the result is still useful as a guide to the design of further studies.

Our work highlighted some important issues for further work in relation to the data as well as the methodology behind their collection and analysis. Apart from the usual problems of noise, the small size and sparsity of the data prevents the algorithm extract-

ing unequivocal rules. In addition, we identified the potential bias introduced when generating trees employing only the cases where factor changes are reported. Therefore, as discussed in section 4.1 we also include the 'no change' instances as part of the training data. In similar work [27], involving trees derived from human-human interactions, trees were aggregated from two different sets of data performing a 'manual' bootstrapping or bagging [7]. A similar approach can also be followed here to enrich the final set of rules.

However, there is a challenge introduced by the assumption that 'no report' is the same as 'no change'. This is not always true. In the case of this study, due to the way we conducted the walkthroughs we were able to prompt students in places where (based on previous and pilot studies) we were interested in particular patterns. For some of the factors, the lack of data in these cases seems not to affect our results. Still, the new patterns that have arisen have to be revisited in future studies. For example, for confidence only 9.65% of the patterns that occur in the logs are without a student report at some point during the pattern history. However, for confusion this percentage rises to 13.5%. It is interesting to observe that 91.5% of the patterns which are not associated with any report, are from students with low or very low previous ability and low post test scores. In addition, the majority of 'missing' values is for the confusion or interest factor. This confirms the intuitive fact that lower ability students are not always good in metacomprehension neither insight into their emotions (see also [1]) and hence they report some factors (especially confusion) less accurately. Having a mechanism that would help us design appropriate circumstances in the studies and prompt them during walkthroughs to provide factor values would be very useful.

A related issue is that our analysis and results depend on students' self reported data, the limitations of which are well discussed in the literature [11,29]. We found that this is particularly the case with the factors that have negative connotations. Students may be "embarrassed" to report values that will make them look lazy (like low effort or boredom) but are keen to report that they are exerting effort. On the positive side, some of our results are comparable to trees generated from a similar approach that looks into predicting students' affective states from analysis of human-human interactions and judgements of tutor's walkthroughs rather than student ones [27]. Students seem a more valid source of evidence for reporting their own emotions and other factors like confidence but tutors seem a more suitable source for the way boredom or effort is perceived. By comparing results from different studies we can, in the future, eliminate the bias introduced by using the self-reports. Aggregating the different trees occurring will help us derive more precise rules. In addition, since the evaluations performed influence our judgements on the validity of the results, the 'blind' stratified cross-validation which we use could benefit from being performed in a way that takes into account that we are dealing with educational data. A more formal process could validate rules against test data comprising of judgements from both expert tutors and students themselves Further work will look into adapting the existing algorithms to deal with such issues. Finally, in relation to implementation, as also described in [12], detection accuracies could be increased by implementing hybrid models which combine rules such as the above with other frameworks (e.g. decision theoretic ones [9]) or more intrusive technologies [25,14].

Acknowledgements

We would like to thank the anonymous conference reviewers, as well as Ryan Baker, Evia Kainada for their valuable comments in previous versions of this paper and particularly Kaśka Porayska-Pomsta for feedback and support during parts of this research.

References

[1] V. Aleven, E. Stahl, S. Schworm, F. Fischer, and R. Wallace. Help-seeking and help design in interactive learning environments. *Review of Educational Research*, 77(3):277–320, 2003.

[2] J. R. Anderson, A. T. Corbett, K. R. Koedinger, and R. Pelletier. Cognitive tutors: Lessons learned. *The Journal of the Learning Sciences*, 4(2):167–207, 1995.

[3] J.R. Anderson. *Rules of the mind*. Hillsdale, NJ: Erlbaum, 1993.

[4] I. Arroyo and B. Woolf. Inferring learning and attitudes with a bayesian network of log files data. In Proceedings of AIED 2005, pages 88–95, 2005.

[5] R.S. Baker, A.T. Corbett, and K.R. Koedinger. Detecting student misuse of intelligent tutoring systems. In *Proceedings of the 7th International Conference on ITS*, 2004.

[6] R.S. Baker, I. Roll, A.T. Corbett, and K.R. Koedinger. Do performance goals lead students to game the system? In Proceedings of AIED 2005, pages 57–64, 2005.

[7] L Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[8] J. Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, pages 37–46, 1960.

[9] C. Conati. Probabilistic assessment of user's emotions in educational games. *Journal of Applied Artificial Intelligence*, 16(7-8):555–575, 2002.

[10] C. Conati, S. Marsella, and A. Paiva. Affective interactions: the computer in the affective loop. In *IUI 2005*. ACM Press, 2005.

[11] A. de Vicente and H. Pain. Informing the detection of students' motivational state: An empirical study. In S.A.Cerri, G. Gouarderes, and F. Paraguaçu, editors, *ITS 2002*, pages 955–962. Springer., 2002.

[12] S. K. D'Mello, S.D. Craig, J. Sullins, and A. C. Graesser. Predicting affective states expressed through an emote-aloud procedure from autotutor's mixed-initiative dialogue. *International Journal of Artificial Intelligence in Education*, 16:3–28, 2006.

[13] S. Draper, M. Brown, E. Edgerton, F. Henderson, E. McAteer, E. Smith, and H. Watt. Observing and measuring the performance of educational technology, TLTP project. 1994.

[14] R. Fernandez, J. Scheirer, and R. Picard. Expression glasses a wearable device for facial expression recognition. Technical report, MIT Media Lab, 1999.

[15] B Kort, R Reilly, and R.W. Picard. An affective model of interplay between emotions and learning: Reengineering educational pedagogy; building a learning companion.

[16] D. Litman and K. Forbes-Riley. Recognizing student emotions and attitudes on the basis of utterances in spoken tutoring dialogues with both human and computer tutors. *Speech Communication*, 48, 2006.

[17] S. Lohr. *Sampling: Design and Analysis*. Duxbury Press, 1999.

[18] M. Mavrikis. Logging, replaying and analysing students' interactions in a web-based ILE to improve student modelling. In Supplement Proceedings of AIED 2005, 2005.

[19] M. Mavrikis and A. Maciocia. WaLLiS: a web-based ILE for science and engineering students studying mathematics. *Workshop of Advanced Technologies for Mathematics Education* in AIED, 2003.

[20] S.K. Mello, S.D. Craig, C.J. Sullins, and A.C Graesser. Predicting affective states expressed through an emote-aloud procedure from autotutor's mixed initiative dialogue. *IJAIED*, 16(1), 2006.

[21] R. Morales, N. van Labeke, and P. Brna. Approximate modelling of the multi-dimensional learner. In *Proceedings of ITS 2006*, pages 555–564, 2006.

[22] Balacheff N. and Sutherland R. Epistemological domain of validity of microworlds, the case of logo and cabri-geometre. In R. Lewis and Mendelshon P., editors, *Proceedings of the IFIP TC3/WG3.3*.

[23] P. Op't Eynde, E. De Corte, and L. Verschaffel. What to learn from what we feel? the role of students' emotions in the mathematics classroom. In S. Volet and S. Jarvela, editors, *Motivation in Learning Contexts. Theoretical Advances and Methodological Implications*. Oxford, UK, 2001.

[24] R.W. Picard. *Affective Computing*. MIT Press, 1997.

[25] R.W Picard and J. Scheirer. The galvactivator: A glove that senses and communicates skin conductivity. In *Proceedings of the 9th International Conference on HCI*.

[26] K. Porayska-Pomsta. *Influence of Situational Context on Language Production: Modelling Teachers' Corrective Responses*. PhD thesis, The University of Edinburgh, 2003.

[27] K. Porayska-Pomsta, M. Mavrikis, and H. Pain. Diagnosing and Acting on Student Affect: the Tutor's Perspective *Submitted to User Modeling and User-Adapted Interaction*, 2006.

[28] R. Stevens, A. Soller, M. Cooper, and M. Sprang. Modeling the development of problem solving skills in chemistry with a web-based tutor. In *Proceedings of the ITS 2004, Maceió, Brasil*, 2004.

[29] A. Stone, C. Bachrach, J. Jobe, H. Kurtzman, and Howard S. *The Science of Self-report; Implications for Research and Practice*. Lawrence Erlbaum Associates, 1992.

[30] I. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques*.

[31] Wood, H. Help seeking, learning and contingent tutoring. *Computers & Education*, 153–169, 1999.

Metacognition

This page intentionally left blank

The Dynamic Nature of Self-Regulatory Behavior in Self-Regulated Learning and Externally-Regulated Learning Episodes

Amy WITHERSPOON [a, c], Roger AZEVEDO[a], Jeffrey GREENE [b], Daniel MOOS [b] and Shanna BAKER [a]
[a]*University of Memphis, Department of Psychology, Institute for Intelligent Systems, Cognition and Technology Lab, Memphis, TN, USA, 38152*
[b] *University of Maryland, College Park, MD, USA, 20742*
[c]*awthrspn@mail.psyc.memphis.edu*

Abstract. This study empirically examines the temporal nature of students' self-regulatory behavior while learning about a complex science topic using hypermedia. The experiment involved randomly assigning 74 undergraduate students to one of two tutoring conditions: self-regulated learning (SRL) or externally-regulated learning (ERL). Participants in the self-regulated learning condition used hypermedia environment to learn about the circulatory system on their own, while participants in the externally-regulated learning condition also used the hypermedia environment, but were given prompts and feedback from a human tutor during the session to facilitate their self-regulatory behavior. Results from product data (mental model pretest-posttest shifts) indicate that the ERL condition leads to a greater likelihood of having a high-level mental model at posttest. In addition, results from the process (think-aloud) data indicate that having access to a human tutor during learning affects the deployment of certain SRL classes at different times within a learning session. Implications for the design of a computer-based learning environment which is intended to stimulate learners' effective use of self-regulated learning are discussed.

Keywords. Self-regulated learning, hypermedia, human tutoring, temporal analysis

Introduction

The potential for students to build deep conceptual understanding of science topics using hypermedia can be limited by learners' ability to utilize self-regulated learning (SRL) processes such as setting and working towards suitable learning goals, monitoring their varying level of understanding, and deploying effective learning strategies [1]. Models of SRL propose that the best way for students to accomplish these objectives is by assessing the learning situation in terms of contextual conditions, setting meaningful learning goals, actively selecting which learning strategies to use, assessing their emerging understanding of the content, deciding how effective previously selected strategies have been in meeting the learning goals, and revamping goals and strategies based on these assessments [2,3,4]. The present study examines the effect of self-regulated learning (SRL) and externally-regulated learning (ERL) on students' mental model shifts and the dynamic nature of the use of SRL processes throughout the SRL and ERL sessions.

It has been suggested that a critical aspect of self-regulated learning demanding more attention in scientific inquiry is how SRL unfolds temporally within learning sessions [5]. To date, we know of no empirical studies investigating the temporal nature of SRL throughout lengthy learning episodes. Winne and colleagues [3,5] propose that students engage in three necessary phases, and a fourth, optional phase during SRL: 1) defining the learning task, including the task and cognitive conditions that exist; 2) setting goals and planning how to accomplish them, 3) enacting tactics (strategies), and 4) adapting metacognition (optional). Winne proposes that these phases are enacted in a cyclical manner throughout a learning session and that there are several feedback loops that provide guidance to the performance of each phase [3]. One approach to investigating the sequence learners follow when attempting to learn with hypermedia is comparing the dynamic patterns of SRL activities used by students learning alone (SRL) with those used by students learning with an external regulating agent (i.e. human tutor; ERL).

In Winne's model [3], a tutor that provides scaffolds to students during learning would be considered a part of the task conditions, which are assessed during the first phase of defining the learning task. However, a tutor can provide students with feedback which would inform students' performance of any phase of the model. For example, a tutor can provide a learner with prompts to activate prior knowledge (Phase 1), set appropriate subgoals (Phase 2), or employ a particular strategy (Phase 3). Also, if a tutor recognizes that a particular strategy that is being used by a student is incongruent with the learner's current goals, the tutor can provide feedback to the student about the ineffectuality of that particular strategy, which would aid the learner in his/her adaptation of metacognition (phase 4). The present study attempts to explore two research questions—*1) Do different tutoring conditions influence learners' ability to shift to more sophisticated mental models of the circulatory system? 2) How do different tutoring conditions influence the deployment of SRL processes at different time periods within a learning session?*

1. Method

1.1 Participants.

Participants included 74 undergraduate non-biology majors from a large public mid-Atlantic university in the United States. The mean age of the participants was 21 years.

1.2. Paper and Pencil Materials.

Materials included a participant informed consent form, participant demographic questionnaire, and identical circulatory system pretest and posttest. The circulatory system pretest and posttest were identical to those used by Azevedo and colleagues (e.g. [6]). The mental model essay pretest was accompanied by the following instruction to participants: "*Please write down everything you can about the circulatory system. Be sure to include all the parts and their purpose, explain how they work both individually and together, and also explain how they contribute to the healthy functioning of the body*."

1.3 Hypermedia Learning Environment (HLE).

The learning session consisted of participants interacting with a commercially-based hypermedia learning environment to learn about the circulatory system. The main relevant articles, which were indicated to the participants during a training phase, were 'circulatory system', 'blood', and 'heart'. These three articles contained 16,900 words, 18 sections, 107 hyperlinks, and 35 illustrations. All of the features of the system, including the search functions, hyperlinks, table of contents, multiple representations (e.g. pictures, videos, etc.) were available to the participants and they were allowed to navigate freely within the environment to any article or representation.

1.4. Procedure.

All participants were tested individually in every condition and participants in the ERL condition were tutored by an individual separate from the experimenter. Participants were randomly assigned to either the SRL (n = 37) or ERL condition (n = 37). Participants were given 20 minutes to complete the mental model essay pretest and immediately given the learning task by the experimenter. Participants in both the SRL group and the ERL group received the following instruction verbally from the experimenter and in writing on a sheet of paper that was available throughout the learning session:

You are being presented with a hypermedia learning environment, which contains textual information, static diagrams, and a digitized video clip of the circulatory system. We are trying to learn more about how students use hypermedia environments to learn about the circulatory system. <u>Your task is to learn all you can about the circulatory system in 40 minutes. Make sure you learn about the different parts and their purpose, how they work both individually and together, and how they support the human body.</u> We ask you to 'think aloud' continuously while you use the hypermedia environment to learn about the circulatory system. I'll be here in case anything goes wrong with the computer or the equipment. Please remember that it is very important to say everything that you are thinking while you are working on this task.

Participants in the ERL condition, in addition to receiving this instruction, had access to a human tutor who scaffolded student's self-regulated learning by:
(1) prompting participants to activate their prior knowledge (PKA);
(2) prompting participants to create plans and goals for their learning and to monitor the progress they were making toward the goals, and
(3) prompting participants to deploy several key self-regulated learning strategies, including summarizing, coordination of informational sources, hypothesizing, drawing, and using mnemonics.

A tutoring script was used by the human tutor in the ERL condition to guide decision making in when prompts should be used and what kind of prompts to implement, given the current status of the learner. The script was created based on literature on human tutoring [7,8] and recent findings from empirical studies on SRL and hypermedia [9,10,11]. For more information about the tutoring script, please see [12].

Think-aloud protocols [13] were collected from all participants during interactions with the hypermedia environment. Participants in both conditions were reminded to continue verbalizing their thoughts by an experimenter who was always nearby, if they were silent for more than three seconds (e.g., "*Say what you are thinking*").

Participants were also reminded of the overall learning goal when they were given the instructions on learning about the circulatory system ("*Make sure you learn about the different parts and their purpose, how they work both individually and together, and how they support the human body*"). The learning session lasted 40 minutes for all participants and the only difference between the SRL and ERL condition was the presence of and tutoring support provided by the human tutor in the ERL condition. All participants had access to paper and pencil and instructed that they could take notes or draw at any point during the learning session, but that their notes would not be available to them during the posttest. Immediately after the 40 minute learning session, participants were given 20 minutes to complete the posttest, which was identical to the pretest mental model essay.

1.5 Coding and scoring of product and process data.

This section describes the scoring procedure used for participants' mental model essays as well as the segmentation used and coding procedure for the think aloud protocols collected from participants.

Mental models. Mental models pretests and posttests were analyzed in order to capture shifts in participant mental models, using qualitative information, based on Azevedo and colleagues' [9,10,11] method of scoring mental models of the circulatory system, (see also research by Chi and colleagues [14,15,16]) was used to analyze participants' conceptual knowledge at pretest and posttest. Participants' initial mental models (pretest) as well as their final mental models (posttest) were classified as belonging to one of 12 categories ranging from (1) no understanding, to (12) an advanced double loop model. A detailed description of necessary components of essays for each category can be found in [9, p. 534-535]. In this study, we re-categorized the original 12 mental models into three qualitatively different mental model categories: *low*, *intermediate*, and *high*. A participant was classified as being in the *low* mental model pretest group if the pretest essay was categorized from (1) to (6) in the initial scoring phase. A participant was classified as being in the *intermediate* mental model pretest group if the pretest essay was categorized as (7) or (8) in the initial scoring phase. If the pretest essay was categorized as (9) or higher, the participant was categorized as being in the *high* mental model pretest group. This same categorization was used for each participant's posttest essay. Each participant's pretest and posttest mental model category (*low*, *medium*, or *high*) was used in the analysis of qualitative shifts in mental models (See [11] for details).

Learners' think-aloud protocols and regulatory behavior. The raw data collected from this study consisted of 2,960 minutes (49.3 hours) of audio and video tape recordings from 74 participants, who gave extensive verbalizations while they learned about the circulatory system. During the first phase of data analysis, a graduate student transcribed the audio tapes and created a text file for each participant. This resulted in a corpus of 1,366 single-spaced pages (M = 18.5 pages per participant) and a total of 378,001 words (M = 5,108.1 words per participant). Azevedo and colleagues [10] developed a coding scheme for SRL processes which was fine-tuned to be used in other studies [17,18] and the later, modified scheme was used to code participants' verbalizations in this study. Several recent theoretical models of SRL [2,3,4] informed the coding scheme's use of the following five classes of variables: *planning, monitoring, strategy use, handling task difficulty and demands,* and *interest*. *Planning* involves creating approaches to learning and setting goals, activating one's knowledge

of the task and contextual conditions, and activating information about one's self in relation to the task and context. *Monitoring* involves metacognitive awareness of the dynamic conditions of the task, context, and self. *Strategy use* involves active deployment of learning strategies in pursuit of a learning goal. *Handling task difficulty and demands* includes efforts to control and regulate different aspects of the task and context. Finally, *interest* includes statements about learner interest in the task or content domain. Descriptions of each SRL variable, the class that each variable belongs to, and examples of each variable can be found in [9, p. 533-534].

Azevedo & colleagues' SRL model was used to segment the think-aloud data for coding corresponding SRL variables to each segment. This resulted in 11,182 segments ($M = 151.1$ per participant) to be coded in the next phase. A coder trained on Azevedo and colleagues' coding scheme then assigned a single SRL variable to each coded segment.

Temporal analysis from the Think-Alouds. In order to perform the time analysis on the SRL classes used by the SRL group and ERL group participants, we divided the original coded transcriptions and into four time intervals. Three trained research assistants watched the video recordings of the learning sessions and noted on each coded transcript the beginning and end of each time interval. The transcriptions were divided into the following four time intervals: 0-10 mins., 10-20 mins., 20-30 mins., and 30-40 mins.. The sequence in which the SRL activities were coded on the transcripts, and the time interval in which they occurred, were entered into a spreadsheet for tallying. Each SRL variable was then categorized as one of the five SRL classes (*planning*, *monitoring*, *strategy use*, *handling task difficulties and demands*, and *interest*) according to the associated class for each strategy [See 9, p. 533-534]. The sequence of SRL class usages in each of four time intervals were calculated for *participant* regulatory behavior only (6,187 segments).

2. Results

2.1 Question 1: Do different tutoring conditions influence learners' ability to shift to more sophisticated mental models of the circulatory system?

Because participants' pretest and posttest mental model groups comprised ordinal (*low, medium, high*) data, ordinal logistic regression was used to identify the influence of both pretest mental model group and condition (SRL and ERL) on participants' mental model posttest group. In these analyses, high mental model posttest group served as the reference, yielding two prediction equations. One equation predicts the odds of being in the low mental model posttest group versus being in the other two groups (intermediate or high) and the other predicts the odds of being in either the low or intermediate group versus being in the high group.

Although a model was run including pretest as a predictor, in addition to condition, the analysis revealed that pretest was not a statistically significant predictor of posttest group ($p > .4$). Following this analysis, another model was run, using only the condition as a predictor. This model provided a statistically significant improvement over the intercept-only model ($\chi^2[1,74] = 4.78$, $p < .05$). The Nagelkerke Pseudo R^2 of this model was .078. The assumption for parallel lines, a necessity in ordinal regression, was met.

Condition was a statistically significant predictor of mental model posttest group in the model (See Table 1). Being in the ERL condition made the odds of being in the high mental model posttest group a little over two times what they were for participants in the SRL condition ($e^{1.118} = 2.38$, so odds were increased by 238%).

Table 1. Ordinal regression predicting mental model posttest group.

	Final Model		
	b	SE (*b*)	e^b
Dependent measure			
Low	-2.033[a]	.451	.13
Intermediate	-1.428[a]	.415	.24
Independent measure			
ERL[c]	1.118[b]	.526	2.38
Model fit			
Final -2 log likelihood	13.746		
Model chi-square/df	4.780/1[b]		
Nagelkerke Pseudo R^2	.078		

[a] $p < .005$ [b] $p < .05$ [c] ERL condition was coded as 1, SRL condition as 0

2.2 Question 2: How do different tutoring conditions influence the deployment of SRL processes at different time periods within a learning session?

This section presents results from a series of 4 X 2 repeated measures ANOVAs on the frequency of SRL class use in each time interval for the two tutoring conditions. A separate 4 X 2 (time interval by tutoring condition) repeated measures ANOVA was run for each SRL class (*planning*, *monitoring*, *strategy use*, *handling task difficulties and demands*, and *interest*) to determine if there were statistically significant differences in the usage of SRL classes across time for the two tutoring conditions.

Planning. A repeated measures ANOVA on the raw frequncy of planning processes across time revealed a statistically significant main effect for time across both tutoring conditions, F (3, 71) = 10.557, p <.001, as well as a statistically significant interaction for time between the SRL condition and the ERL condition, F (3, 71) = 10.950, p < .001. Post-hoc analysis revealed that, when compared to the first, second, and third time interval, participants in both the SRL and ERL condition used statistically significantly more planning activities in the fourth time interval (p < .001). Participants in the ERL condition also used statistically significantly more planning processes in the fourth time interval (p < .001), when compared to the SRL condition. There were no significant differences for any of the remaining comparisons.

Strategy Use. A repeated measures ANOVA on the raw frequency of strategy use across time revealed a statistically significant main effect for time across both tutoring conditions, F (3, 71) = 8.486, p < .001. Post-hoc analysis revealed that, when compared to the third time interval, participants in both conditions used statistically significantly more strategies in the first time interval (p < .001) and the second time interval (p < .001). Also, when compared to the fourth time interval, participants used statistically significantly more strategies in the first time interval (p < .01) and the second time interval (p < .01). There were no significant differences for any of the remaining comparisons and the results did not demonstrate statistically significant interaction between time and condition.

There were no statistically significant differences found for monitoring use, handling task difficulty and demands, or interest across the time intervals.

3. Implications for the design of a computerized SRL tutor

Results indicate that not only do learners deploy differential amounts of planning, strategy use, and interest processes at different periods throughout a learning session, but also that a human tutor can affect the rate at which these activities are deployed at different times. In addition, learners in the ERL condition demonstrated higher odds of having a higher mental model at posttest than those in the SRL condition. This has implications for the design of a computerized tutor developed to foster students' learning about complex science topics. A computerized tutor should emulate the contributions from the human tutor in the ERL condition. A human tutor, using fairly strict guidelines in prompting students to engage in these activities, proves effective at eliciting the use of certain SRL activities at different times throughout the session. This suggests that a computerized tutor, with capabilities to track students' level of understanding, could provide effectual prompts to students as well.

In general, results indicate that learners use a larger amount of planning activities in the later part of their learning session. In contrast, learners appear to use a greater amount of strategies at the beginning of the learning session than in the later part. This could be due, in large part, to participants' overall low prior domain knowledge of the circulatory system. Learners may experience difficulty planning in a topic they have very little background in, prior to exposure to the hypermedia environment. Instead, learners attempt to use several strategies (e.g. summarizing, re-reading, taking notes) they believe might help them develop their knowledge about the topic.

As the learning session progresses, and learners acquire knowledge of the topic, they become more capable of using planning activities such as prior knowledge activation. Analogously, learners probably are unlikely to demonstrate much interest in a domain they have very little to no background in, but once experience with the hypermedia environment provides the learners with information about the complex science topic, they will demonstrate greater interest.

These findings have implications for the design of an adaptive hypermedia system intended to adaptively respond to a learner's developing understanding of a domain. One design principle which these findings suggest is that adaptive environments include the built-in assumption that learners will probably use more planning and interest activities toward the later portions of the session. In addition, an adaptive CBLE can capitalize on learner's developing understanding by prompting the use of planning and strategy use more often toward the end of the learning session. Although in this experiment it was shown that learners use strategies more often toward the beginning of the learning session, it might be preferable that the deployment of effective strategies, for example, coordination of informational sources [12]. In regard to the observed rise in interest later in learning sessions, more research should be conducted to investigate the motivational aspects of computer-based learning environments [e.g. 18], especially in connection with the deployment of self-regulated learning activities.

Acknowledgements

This research was supported by funding from the National Science Foundation (REC#0133346 and REC#0633918) awarded to the second author.

References

[1] Azevedo, R. (2005). Using hypermedia as a metacognitive tool for enhancing student learning? The role of self-regulated learning. *Educational Psychologist, 40*(4), 199-209.

[2] Pintrich, P. R. (2000). The role of goal orientation in self-regulated learning. In M. Boekaerts, P. Pintrich, & M. Zeidner (Eds.). *Handbook of self-regulation* (pp. 451-502). San Diego, CA: Academic Press.

[3] Winne, P. H. (2001). Self-regulated learning viewed from models of information processing. In B. Zimmerman & D. Schunk (Eds.), *Self-regulated learning and academic achievement: Theoretical perspectives* (pp. 153-189). Mahwah, NJ: Erlbaum.

[4] Zimmerman, B. (2000). Attaining self-regulation: A social cognitive perspective. In M. Boekaerts, P. Pintrich, & M. Zeidner (Eds.), *Handbook of self-regulation* (pp. 13-39). San Diego, CA: Academic Press.

[5] Winne, P. H., & Perry, N. E. (2000). Measuring self-regulated learning. In M. Boekaerts, P. Pintrich, & M. Zeidner (Eds.). *Handbook of self-regulation* (pp. 531-566). San Diego, CA: Academic Press.

[6] Azevedo, R., Cromley, J. G., Winters, F. I., Moos, D. C., & Greene, J. A. (2005). Adaptive human tutoring facilitates adolescents' self-regulated learning with hypermedia. *Instructional Science, 33*, 381-412.

[7] Chi, M. T. H. (1996). Constructing self-explanations and scaffolded explanations in tutoring. *Applied Cognitive Psychology, 10*, S33-S49.

[8] Graesser, A. C., Person, N. K., & Magliano, J. P. (1995). Collaborative dialogue patterns in naturalistic one-to-one tutoring. *Applied Cognitive Psychology, 9*, 495-522.

[9] Azevedo, R., & Cromley, J. G. (2004). Does training on self-regulated learning facilitate students' learning with hypermedia? *Journal of Educational Psychology, 96*(3), 523-535.

[10] Azevedo, R., Guthrie, J. T., & Siebert, D. (2004). The role of self-regulated learning in fostering students' conceptual understanding of complex systems with hypermedia. *Journal of Educational Computing Research, 30*(1), 87-111.

[11] Azevedo, R., Cromley, J. G., & Seibert, D. (2004). Does adaptive scaffolding facilitate students' ability to regulate their learning with hypermedia? *Contemporary Educational Psychology, 29*, 344-370.

[18] Azevedo, R., Moos, D.C., Greene, J.A., Winters, F.I., & Cromley, J.C. (in press). Why is externally-regulated learning more effective than self -regulated learning with hypermedia? *Educational Technology Research and Development*.

[13] Ericsson, K. A. & Simon, H. (1993). *Protocol analysis: Verbal reports as data*. Cambridge, MA: MIT Press

[14] Chi, M. T. H., de Leeuw, N., Chiu, M. H., & LaVancher, C. (1994). Eliciting self-explanations improves understanding. *Cognitive Science, 18*, 439-477.

[15] Chi, M. T. H., Siler, S., & Jeong, H. (2004). Can tutors monitor students' understanding accurately? *Cognition and Instruction, 22*, 363-387.

[16] Chi, M. T. H., Siler, S., Jeong, H., Yamauchi, T., & Hausmann, R. (2001). Learning from human tutoring. *Cognitive Science, 25*, 471-534.

[17] Moos, D. C., & Azevedo, R. (2006). The role of goal structure in undergraduates' use of self-regulatory variables in two hypermedia learning tasks. *Journal of Educational Multimedia and Hypermedia, 15*(1), 49-86.

[18] Moos, D. C., & Azevedo, R. (in press). Exploring the fluctuation of motivation and use of self-regulatory processes during learning with hypermedia. *Instructional Science.*

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

The Influence of External-Regulation on Student Generated Questions during Hypermedia Learning

Jeremiah SULLINS, Roger AZEVEDO, Andrew TROUSDALE and Jennifer SCOTT
University of Memphis, Dept. of Psychology, Institute for Intelligent Systems,
3693 Norriswood Avenue, Memphis, TN, 38152, USA
jsullins@memphis.edu

Abstract. The study examined the differences in the frequency and quality of student generated questions while using a hypermedia environment. We also examined what specific self-regulatory processes were present prior and following student generated questions. The experiment involved 20 college students from a larger study and categorized them into two separated groups. One group, which was called low shifters, contained participants who had mental model shifts from one to five out of a possible twelve, regardless of whether they started with low or high mental models. The second group, which was called high shifters, contained participants who had mental model shifts above five out of a possible twelve, regardless of whether they started with low or high mental models. Analysis revealed that there were no significant differences in the raw number of questions asked between the participants in the low shifters group and those in the high shifters group. Additional analysis also found that there were no significant differences in the amount of "deep" or "shallow" questions asked by participants in the two groups. In addition, we discovered that there was a significant difference in the specific self-regulatory processes that were used by the high shifters following the generation of a question. More specifically, the role of metacognitive processes is key to understanding the nature of question-generation during tutoring. We propose several implications for the design of intelligent learning environments (ILEs).

Keywords. Self-regulated learning, question asking, hypermedia, interactive learning environments

Introduction

Do the types of questions generated by students while being tutored about a complex science topic with hypermedia influence learning and self-regulatory processes? There is some dispute among psychologists that question generation is an integral part in such things as comprehension [1, 2, 3], problem solving, and reasoning [4, 5]. In addition, asking good questions has been shown to lead to improved comprehension, learning, and memory of the materials among school children [6, 7, 8, 9, 10]. Although research has shown help-seeking behavior, such as question asking, to be a crucial metacognitive skill, question generation by students is infrequent in traditional settings, such as a classroom [11, 12, 13]. One possible reason that students perform more efficiently in tutoring sessions is the quality of the questions that the tutor directs towards the student. It has been shown that only a small percentage of teacher-generated questions in the classroom are high-level questions (4%). Most of the questions directed to the students are shallow level questions explicitly designed to test the students' explicit memory.

Classroom question generation between a teacher and students undoubtedly plays a major role in the domain of help seeking behaviors; however, an increasing number of researchers are becoming interested in the role that technology plays [14, 15]. As educational technology becomes available to a larger proportion of the population, more people have access to educational tools, such as interactive learning environments (ILEs). ILEs are computerized learning environments, such as intelligent tutoring systems or hypermedia environments, that are targeted at students and provide on-demand hints, hyperlinks, and online glossaries when needed by the learners. ILEs, which are becoming dominant in educational practice and learning research, are designed to address the needs of individual learners' by assisting them with help facilities. Unfortunately, research is beginning to show that learners' are not taking advantage of this opportunity because they either abuse the help function, or do not use the help function when it is appropriate [16, 17, 18]. Given the increase of technology in education and the fact that learners are abusing the help functions of these technologies, it is beneficial to study the effects of the presence of a human tutor as a facilitator during the use of a hypermedia learning environment (e.g., [17]).

In our research, we examine learning from a self-regulation perspective [19]. SRL is an active, constructive process whereby learners set goals for their learning and then attempt to monitor, regulate, and control their cognition, motivation, and behavior in the service of those goals [19, 20, 21]. SRL is guided and constrained by both personal characteristics and the contextual features of the environment [22]. Previous research has hypothesized the importance of question generation in traditional environments such as classrooms and one-to-one tutoring, but we are interested in the novel idea of what role question generation plays while students are self-regulating their learning about a complex science topic with hypermedia. Traditionally, we have examined shifts in participants' mental models, based on pretests and posttests, to assess the amount of learning that is occurring with a hypermedia environment [23]. The present study examines the influence of question asking and external-regulation on shifts in participants' mental models. To determine if there is a difference in self-regulation and question asking, we examined two groups of participants, those who have low mental model shifts (participants who had mental model shifts of one to five, regardless of pretest mental model at start of task) and those who exhibited high mental model shifts (participants who had mental model shifts of more than five, regardless of pretest mental model at start of task). More specifically, in this paper we focus on three research questions—1) *Does the frequency of question asking differ between participants who had high mental model shifts versus participants who had low mental model shifts?* 2) *Do participants who show high mental model shifts ask more deep reasoning questions than participants who show low mental model shifts?* 3) *What specific self-regulatory processes are being used at the time preceding and following a participant generated question?*

2. **Method**

2.1 Sample.

Participants in this study were a subset from a larger study from the fall of 2003 and spring of 2004 from a large public university who received extra credit in their educational psychology course for their participation [24]. The students were non-biology majors, and the pretest results demonstrated that all participants had average or little knowledge of the circulatory system. Out of a sample of 41 in the tutoring condition, we selected 20 participants to determine if there was a significant difference in the frequency and quality of the questions generated by the participants who exhibited high shifts in their mental models versus the participants who showed low shifts in their mental models. In the present study, the 20 participants from the tutoring condition were broke into two separate groups. The first groups, called low shifters, were those participants that scored a one to five out of 12 on the mental model shift from pretest to posttest. The second group, called high shifters, were those participants that scored a six or higher out of 12 on the mental model shift from pretest to posttest (see below for a description of the mental model categories).

2.2 Measures.

The materials consisted of a paper and pencil consent form, participant questionnaire, pretest, and posttest. The materials were the same as those developed and used by Azevedo and colleagues [24, 25, 26]. The pretest consisted on a mental model essay asking the participant to write down everything they knew about the circulatory system. On the mental model pretest, the participant could obtain a total of 12. Our mental models are based on Chi's previous work with mental models [27, 28] (see [29] for details).

2.3 Experimental Procedure.

As part of the larger study, participants were randomly assigned to one of two experimental conditions. Participants in the control condition interacted with the hypermedia environment for 40 minutes with no assistance. Those participants in the human tutoring condition interacted with a hypermedia environment, but had the option to relay any content related questions they had to a human tutor during the learning session. All participants in both groups were told to "think aloud" during the entirety of the session. An experimenter was present at all times to remind the participants to verbalize if they had been silent for more than three seconds.

2.4 Hypermedia Learning Environment.

During the experimental phase of the study, participants were asked to interact with a hypermedia environment over a 40 minute time span. The commercially-based hypermedia environment contained text, diagrams, hyperlinks, and video. Participants were shown at the beginning of the 40 minutes what the most relevant articles were on the circulatory system (circulatory system, blood, and heart). However, they were told they could explore any part of Encarta they wanted to.

2.5 Coding and Scoring.

Following the categorization of the participants into the two groups, we then used the transcripts to analyze the questions and individual self-regulatory processes. To analyze the questions, an experimenter went through each individual transcript and identified the questions that the participants directed towards the human tutor. The questions were then extracted and classified as either deep or shallow questions based on a question taxonomy by Graesser and Person (1994). The question taxonomy contains 18 question categories. An example of a shallow question, according to the taxonomy, would be a verification question, (e.g., "*Can I just kind of trace this?*"), in which this student is referring to a diagram of the heart. The purpose for categorizing this type question as "shallow" is that it does not take much intrinsic thought on the student's part. It is a simple yes or no question. In contrast, an example of a deep-reasoning question would be an expectational question (e.g., "*From my science classes, but what does it mean, like, so why is it that your arteries can clog bases upon cholesterol level but your veins don't really?*"). The reasons for categorizing this question as "deep" is because the student must take the knowledge he or she knows about the circulatory system, and make a connection and compare the differences between the two. The answers to "deep" questions are usually around a paragraph in length.

To analyze the individual self-regulatory processes that preceded and followed the question, an experimenter then took the transcripts, which had previously been coded [24] for all self-regulatory processes in a previous study [24] and identified which self-regulatory process preceded and followed an individual question. There were a total of 544 self-regulatory processes that either preceded (272) or followed (272) a question. The SRL processes were categorized as planning (e.g., prior knowledge activation, recycle goal in working memory), monitoring (e.g., content evaluation, feeling of knowing), strategy (e.g., coordinating informational sources, draw), task difficulty and demands (e.g., help seeking behavior, task difficulty), interest (e.g., interest statement), or feedback (positive or negative) [29].

3. Results

Question 1: Does the frequency of question asking differ between participants who had high mental model shifts versus participants who had low mental model shifts? To analyze the difference in the number of questions asked by participants in the high mental model shift versus those in the low mental model shift, an independent t-test was used in the analysis. The analysis indicated that there was no statistically significant difference between the raw number of questions generated by the high shifters (M= 15.90) versus the raw number of questions generated by the low shifters (M= 11.40), t (18) = -.962, p = .349. This suggests that both high- and low-shifters generated the same number amount of questions when given access to a human tutor to facilitate their learning about a complex science topic with a hypermedia.

Question 2: Do participants who show high mental model shifts ask more deep reasoning questions than participants who show low mental model shifts? To examine if

there was a difference in the distribution of deep-level questions asked by high shifters versus low shifters, a 2 X 2 (low and high shifters versus shallow and deep-reasoning) chi-square analysis was performed. The analysis showed that there was no statistically significant differences in the distribution of shallow or deep reasoning questions asked by high versus low shifters (χ^2[1, N= 271] = 1.42, p > .05). This suggests that although some research has made the argument that only learners who have a deep comprehension are competent enough to ask deep reasoning questions, our results indicate that given the presence of a human tutor, both low and high shifters are asking approximately the same amount of deep and shallow questions while learning with a hypermedia environment.

Question 3: What self-regulatory processes are being used at the time preceding and following a participant generated question? In this section we present the results of a series of chi-square analyses that were performed to determine if there was a difference in the distribution of self-regulatory processes used *prior* to and *following* the student generated questions. The purpose of examining the SRL processes that occur before and after question generation is that since there was no difference in the raw number of questions or the depth of the question asked, there might be different metacognitive strategies employed by the high shifters. The self-regulatory categories that were examined were planning, monitoring, strategy, task difficulty and demand, and feedback. The distribution of self-regulatory processes that were present during the 40 minutes with the hypermedia environment can be seen in Figure 1.

First we examined to see if there was a difference in the amount of self-regulatory processes that were used *prior* to the learners' questions. A 6 X 2 (self-regulatory processes by high shifters versus low shifters) chi-square revealed that there were statistically significant differences in the amount of SRL processes that were used by high shifters versus the distribution of SRL processes that were used by low shifters preceding question generation (χ^2 [5, N = 272] = 14.99, p < .05. These results suggest that (compared to the low shifters) the high shifters are exhibiting significantly more SRL processes *before* the generation of a question than the participants in the low shifters category.

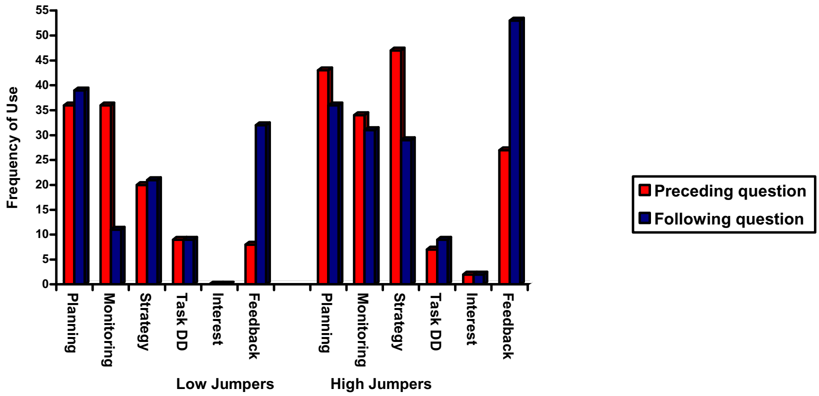
Our next analysis was conducted to see if there was a difference in the amount of SRL processes used by the high shifters and the low shifters *following* question generation. A 6 X 2 (self-regulatory processes by high versus low shifters) chi-square revealed that there were no statistically significant differences between the high and low shifters in the distribution of SRL processes that were used following question generation (χ^2 [5, N = 272] = 9.95, p > .05. These results suggest that neither group of learners differed in the distribution of SRL processes deployed following a question.

Since the chi-square analysis revealed that there was a significant difference between the SRL processes that were used between high shifters and the low shifters, we then examined the distribution of SRL processes by both high and low shifters *prior* to and *following* the questions for each individual SRL category.

Monitoring. Chi-square analysis revealed statistically significant differences in the distribution of monitoring processes used following question generation by either the low shifters or the high shifters (χ^2[1, N= 112] = 6.87, p < .05). After examining the data, we were able to indicate which specific metacognitive monitoring skills were used by both groups during learning. One of the first metacognitive monitoring processes used was tutor

instructed feeling of knowing, which is the tutor prompting the student whether he or she is aware of having read or learned something in the past and have some understanding of the material (e.g., “Have you read that in a biology textbook somewhere before?”). Another monitoring process included judgment of learning, in which is the student stated that they understand a fact or idea that was just presented (e.g., “I get it!” or “that makes sense!”). In addition, we also found that

Figure 1.
SRL Processes that Preceded and Followed Participants' Questions



students also initiated feeling-of-knowing (FOK), when the learner was aware of having read or learning something in the past and having some understanding of it (e.g., “Oh, I already read that.”). Also, the tutor suggested that the learner should monitor their progress toward goals quite frequently during learning. The goal of this tutor-generated metacognitive monitoring process is to bring to the attention of the learner whether a previously set goal has been met (e.g., “Is that something you should be familiar with for the posttest?”), and how many more goals need to be accomplished. Finally, the tutor prompted the student to evaluate the content of the hypermedia environment, which is the idea that the tutor made the student aware that a just seen text, diagram, or video is beneficial to learning (e.g., “that is probably something you should try to remember.”). Additional analyses were performed on the remaining SRL processes but no significant differences were found.

4. **Summary and Results**

Our results suggest that although question generation in previous research is an effective tool in the facilitation of learning [11,15], question generation may not play a crucial role in increasing learning gains when learning about a complex science topic in a hypermedia environment. When it comes to the raw number of questions that both the learners in the high shifters category and the learners in the low shifters category asked, there were no differences between the learners in the two groups. In addition, when looking at the depth of questions asked by both groups of learners, again there were no differences between the

learners in the two groups. Although some research has shown that learners who acquire deep comprehension of materials ask deep reasoning questions, our results indicate that this is not the case when learners are asked to learn about the circulatory system in a hypermedia environment. In addition, the results from the chi-square analyses reveal that both the low and high shifters are deploying metacognitive self-regulatory processes by demonstrating they do not understand the material by exhibiting help-seeking behaviors. According the chi-square analyses, high shifters are showing a higher degree of metacognitive monitoring (i.e., judgment of learning, feeling of knowing).

5. **Implications for Interactive Learning Environments**

After examining the results from the chi-square analyses, we can see that the only difference in self-regulation is the amount of metacognitive monitoring that the high shifters demonstrate at the time following a question. Since we are aware of what self-regulatory processes are beneficial in the facilitation of learning [19, 25], the design of an interactive, adaptive learning environment has the potential to be tailored to the needs of the learner. As mentioned earlier, many students are abusing help functions of ILE's or ignoring them all together. The results from the present study provide evidence for the future direction of ILE's by suggesting what types of self-regulation processes is occurring at the time of student question generation. For example, if a student is engaged in a task with an intelligent tutoring system to learn about a challenging science topic related to a body system, and the student encounters some form of cognitive disequilibrium by self-regulating and monitoring their learning, we will know what form of self-regulation is most beneficial for the learner, and therefore have the ILE prompt the student to deploy effective self-regulatory techniques. In an ideal situation, the student could interact with an interactive learning environment, such as an intelligent tutoring system, and the system could provide prompts and feedback based on natural language processing. As the student encounters some form of cognitive disequilibrium based on their self-regulation and metacognitive monitoring, and types a question to the intelligent tutoring system, the system in real time, can then provide appropriate SRL prompts based on our results (e.g., FOK, JOL).

Acknowledgements

This research was supported by funding from the National Science Foundation (REC#0133346 and Reese#0633918 awarded to the second author.

References

[1] Collins, A., Brown, J., & Larkin, K. (1980). Inference in text understanding. In R. Spiro, B.Bruce, and W. F. Brewer (Eds.), *Theoretical issues in reading comprehension* (pp. 385-407). Hillsdale, NJ: Erlbaum.

[2] Graesser, A. C., Singer, M., & Trabasso, T. (1994). Constructing inferences during narrative text comprehension. *Psychological Review, 3*, 371-395.

[3] Kintsch, W. (1998). *Comprehension: A paradigm for cognition.* NY: Cambridge University Press.

[4] Graesser, A. C., Baggett, W., & Williams, K. (1996). Question-driven explanatory reasoning. *Applied Cognitive Psychology, 10*, S17-S32.

[5] Sternberg, R. J. (1987). Questioning and intelligence. *Questing Exchange, 1*, 11-13.

[6] Davey, B., & McBride, S. (1986). Effects of question generating training on reading comprehension. *Journal of Educational Psychology, 78*, 256-262.
[7] Gavelek, J. R. & Raphael, T. E. (1985). Metacognition, instruction, and the role of questioning activities. In D. L. Forrest-Presslet, G. E. MacKinnin, & T. G. Waller (Eds.), *Metacognition, cognitive, and human performance: Vol. 2* (pp. 103-136). Orlando, Fl: Academic Press.
[8] King, A. (1989). Effects of self-questioning training on college students' comprehension of lectures. *Contemporary Educational Psychology, 14*, 366-381.
[9] King, A., Staffieri, A., & Adelgais, A. (1998). Mutual peer tutoring: Effects of structuring tutorial interaction to scaffold peer learning. *Journal of Educational Psychology, 90*, 134-152.
[10] Palincsar, A. S., & Brown, A. (1984). Reciprocal teaching of comprehension-fostering and comprehension-monitoring activities. *Cognition and Instruction, 1*, 117-175.
[11] Graesser, A. C, & Person, N. (1994). Question asking during tutoring. *AERJ, 31*, 104-137.
[12] Graesser, A.C., McNamara, D.S., & VanLehn, K. (2005). Scaffolding deep comprehension strategies through Point&Query, AutoTutor, and iSTART. *Educational Psychologist, 40,* 225-234.
[13] Cohen, P. A., Kulik, J. A., & Kulik, C. C. (1982). Educational outcomes of tutoring: A meta-analysis of findings. *American Educational Research Journal, 19,* 237-248.
[14] Aleven, V., Stahl, E., Schworm, S., Fischer, F., & Wallace, R.M. (2003). Help Seeking in InteractiveLearning Environments. *Review of Educational Research, 73*(2), 277-320.
[15] Aleven, V., McLaren, B.M., Roll, I., & Koedinger, K.R. (2006). Toward meta-cognitive tutoring: A model of help seeking with a Cognitive Tutor. *IJAIED*, (16), 101-30.
[16] Aleven, V. & Koedinger, K. R. (2000). Limitations of Student Control: Do Student Know when they need help? In G. Gauthier, C. Frasson, and K. VanLehn (Eds.), Proceedings of the 5th International Conference on Intelligent Tutoring Systems, ITS 2000 (pp. 292-303). Berlin: Springer Verlag.
[17] Azevedo, R., Cromley, J., Winters, F., Moos, D., & Greene, J.(2005). Adaptive human scaffolding facilitates adolescents' self-regulated learning with hypermedia. *Instructional Science, 33,* 381-412.
[18] Aleven, V., McLaren, B. M., Roll, I. and Koedinger, K. R. (2004). Toward Tutoring Help Seeking: Applying Cognitive Modeling to Meta-Cognitive Skills; In the Proceedings of the 7th Conference (ITS-2004).
[19] Azevedo, R. (2005 b). Using hypermedia as a metacognitive tool for enhancing student learning? The role of self-regulated learning. *Educational Psychologist, 40(4),* 199-209.
[20] Winne, P.H. (2001). Self-regulated learning viewed from models of information processing. In B. Zimmerman & D. Schunk (Eds.), *Self-regulated learning and academic achievement: Theoretical perspectives* (pp. 153-189). Mawah, NJ: Erlbaum.
[21] Winne, P, & Hadwin, A. (1998). Studying as self-regulated learning. In D.Hacker, J. Dunlosky, & A. Graesser (Eds.), *Metacognition in educational theory and practice* (pp. 277-304). Hillsdale, NJ: Erlbaum.
[22] Zimmerman, B., & Schunk, D. (2001). Theories of self-regulated learning and academic achievement: an overview and analysis. In *Self-regulated learning and academic achievement* (pp. 1-37). Mahwah, NJ; Erlbaum.
[23] Pintrich, P. (2000). The role of goal orientation in self-regulated learning. In M. Boekaerts, P. Pintrich, & M. Zeidner (Eds.), *Handbook of self-regulation* (pp. 451-502). San Diego, CA: Academic Press.
[24] Azevedo, R., Moos, D., Greene, J., Winters, F., & Cromley, J. (in press). Why is externally-regulated learning more effective than self-regulated learning with hypermedia? *Educational Technology Research and Development.*
[25] Azevedo, R., Cromley, J., Winters, F., Moos, D., & Greene, J.(2006). Using computers as metacognitive tools to foster students' self-regulated learning. *TICL Journal, 3,* 97-104.
[26] Greene, J.A., & Azevedo, R. (2005). Adolescents' use of SRL behaviors and their relation to qualitative mental model shifts while using hypermedia. In C-K. Looi, McCalla, G., Bredeweg, B., & Breuker, J. (Eds.), *Artificial intelligence in education: Supporting learning through intelligent and socially informed technology* (pp.233-240). Amsterdam, The Netherlands: IOS Press.
[27] Chi, M. T. H., de Leew, N., Chiu, M., & LaVancher, C. (1994). Eliciting self-explanations improves understanding. *Cognitive Science, 18*, 439-477.
[28] Chi, M. T. H. (2000). Self-explaining: The dual processes of generating inference and repairing mental models. In R. Glaser (Ed.), *Advances in instructional psychology: Educational design and cognitive science* (Vo. 5, pp. 161 -- 238). Mahwah, NJ: Erlbaum.
[29] Azevedo, R., & Cromley, J.G., (2004). Does training on self-regulated learning facilitate students' learning with hypermedia? *Journal of Educational Psychology, 96,* 523-535.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Repairing Disengagement With Non-Invasive Interventions

Ivon ARROYO, Kimberly FERGUSON, Jeff JOHNS, Toby DRAGON, Hasmik MEHERANIAN, Don FISHER, Andrew BARTO, Sridhar MAHADEVAN, Beverly P. WOOLF
Department of Computer Science & Department of Engineering, University of Massachusetts Amherst

Abstract. We evaluated the impact of a set of interventions to repair students' disengagement while solving geometry problems in a tutoring system. We present a deep analysis of how a tutor can remediate a student's disengagement and motivation with self-monitoring feedback. The analysis consists of a between-subjects analyses on students learning and on students' attitudes towards mathematics and perceptions of the software, and a deeper analysis on students' engagement within the tutor. Results show that the general trend over time is for students to become more disengaged while using a tutor, yet students can be coaxed into reengagement after viewing interventions that promote self-reflection and self-monitoring --a simple open-learner model accompanied by suggestions and encouragement.

Introduction

Students become disengaged overtime when using tutoring software. One particular externalization of disengagement is "gaming" the system, or moving rapidly through problems without reading them, quickly moving through hints, and seeking the final hint that might give the answer away. It has been estimated that students who game the system learn two thirds of what students who do not game the system learn (Baker, Corbett & Koedinger, 2004). This could be a sign of frustration, something especially important to detect for students with special needs in particular. Another possibility is that this is a sign of poor use of meta-cognitive resources –the ability to regulate their own cognitive resources (Hartman, 2001). In either case, by identifying disengagement behavior and repairing it we move closer towards tutors that generate highly individualized, pedagogically sound and accessible material. This research shall lead towards involving more students, including those that might be unmotivated, or those that are anxious about the taught domain or about computers in general, or that simply are not aware of the impact of their unproductive actions on the learning process.

Developing pedagogical approaches to respond online to students who have become disengaged is a challenging task. Past research showed that providing students with examples (extra materials to help in the solving of the problem at hand) can reduce gaming and improve learning (Baker et al., 2006). One possible reason for the success of this feedback is that the origin of gaming was due to frustration for not having enough knowledge to solve the problem. It is still unclear though if students can modify their unproductive behaviours and learn more without the provision of extra content help, just by having them reflect on their actions, and making them aware of productive behaviours. This kind of scaffolding is called 'meta-cognitive' feedback because it tackles self-regulatory processing while learning (Hartman, 2001). De Jong and van Joolingen (1998) described ways in which learning environments can support students' metacognitive skills; however, it has still not been demonstrated whether Software Learning Environments can help students

become better learners by providing such meta-cognitive support. Some attempts to remediate unproductive uses of a tutoring software showed that students became even more frustrated when their gaming was blocked with pop-up windows that encouraged proper behavior whenever specific meta-cognitive bugs were observed (Aleven et al., 2004). Not surprisingly, the benefit in actual learning for that intervention was limited (Roll et al, 2006).

One possible approach to making students better learners is to make them more aware of productive and unproductive behaviours, and about the consequences of their actions on their progress, in a way that does not block their unproductive behaviours. With that idea in mind, we designed and evaluated the impact of pedagogical interventions that invite students to reflect on their progress in-between problems, and understand what would be behaviours that will benefit their learning. This paper evaluates the hypothesis that such non-invasive interventions can change a student's engagement state, reduce gaming, enhance learning, while at the same time generate a more positive perception of the system and of the learning experience. More specifically, two hypotheses were tested as part of this research: i) do in-between-problems interventions (performance charts and tips) affect the level of student engagement? and ii) do interventions impact student learning and feelings towards the tutor and towards the learning experience? How about feelings towards their own self-concept and mathematics? The next sections are our attempt to answer these questions.

1. Methodology and Experiment Design

The tutor. Wayang Outpost is a multimedia tutoring system for geometry (Arroyo et al, 2004). It helps students solve challenging geometry standardized tests problems. Wayang is a web-based tutoring software (a database-backed Java servlet with a Macromedia Flash front-end), facilitating the task of logging, pre/post-testing and data collection in general. Students register to use the system, log on and are directed towards the different modules for pre/post-testing and tutoring. Even though Wayang has an adaptive module to tailor the sequencing of problems depending on students' performance at past problems, for this study, the sequencing of problems in Wayang Outpost was fixed (i.e. the same for all students). This decision was made thinking that, if the interventions had an impact, students' engagement in problems would in turn affect problem-solving behavior and interact with the problem sequencing, and the results of this study. Thus, Wayang provided a fixed sequence of problems, chunking problems of similar skills close to each other, and organizing them from easy to hard overall difficulty.

Instruments. Two mathematics tests of 43 items extracted from SAT and Massachusetts MCAS standardized tests (MCAS) were used for pre and post testing. We examined the standardized test scores from 10th graders who took this exam days after the experiment finished. Post-tutor survey question items were provided, including a student's performance/learning orientation, human-like attributes of tutor and a student's liking of mathematics. Most of these came from a Intervention instrument used by Baker (2006) for studies about gaming the system. Items to measure self-concept in mathematics (Wigfield and Karpathian, 1991) and a 10-question self-efficacy instrument were also part of the survey. Last, we included questions that measured student perceptions of the software and the help that we designed. All items were in a 6-likert-type scale, except for the 2 learning vs. performance orientation items (Mueller&Dweck, 1998).

Experimental design. Eighty eight (88) students from four different classes (10th grade students and some 11th graders) from an urban-area school in Massachusetts used Wayang Outpost for one week. It was 4 time periods for about 2 hours of tutoring (the rest of the time was spent doing pre-testing and post-testing). A second control group (called no-tutor control) consisted of matched classes of students who did not use the tutor at all, but were of the same grade level, equivalent proficiency level, and taught by the same teachers. When students logged on to the software, they were pseudo-randomly assigned to either the experimental (Interventions) or the tutor control group. The latter group used the traditional Wayang –worked on problems with the ability to click on a help button that would provide multimedia hints. The Intervention Group received intervention screens at fixed intervals of 6 problems (i.e., after clicking the 'next problem' button on the sixth problem). Experimental interventions were either i) a performance graph with an accompanying message, similar to Figure 1 (students received a negative graph or a positive graph depending on their recent performance and their past performance) or ii) a tip that suggested a productive learning behavior. The tutor provided two kinds of tips: Tip-read-carefully and Tip-make-guess. Tip-read-carefully encouraged students to slow down, read the problem and hints carefully ("Dear Ivon, We think this will make you improve even more: Read the problem thoroughly. If the problem is just too hard, then ask for a hint. Read the hints CAREFULLY. When a hint introduces something that you didn't know, write it down on paper for the next time you need it"). Tip-make-guess encouraged the student to think about the problem, make a guess and, if the guess was wrong, ask for hints ("Dear Ivon, Think the problem thoroughly and make a guess. If your guess is wrong, no problem, just ask for a hint. If you need more hints, keep clicking on help". Students were addressed by their first name both in the messages accompanying the charts and the tips. Whether a student saw a progress chart or a tip, and which one, was a randomly-made decision.

Procedure. During the first time period, students took an online mathematics pretest. Then they used the tutoring software for part of the first day, second and third days. Posttest was started at the end of the third day and continued during the fourth day. Pre-test and post-tests were completed online (within the software). Pretests and posttest were counterbalanced.

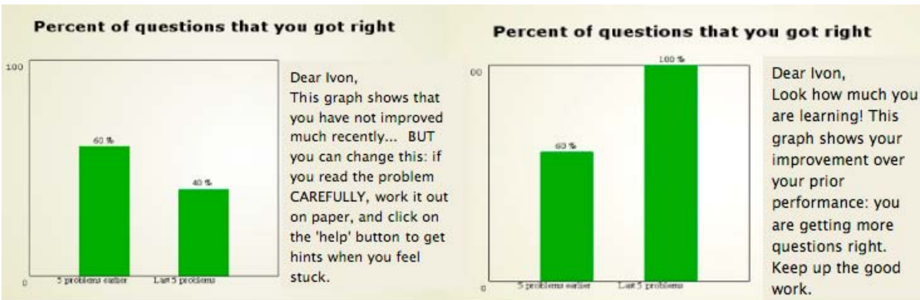


Figure 1. Progress Charts that show students their accuracy of responses from before to recently

2. Results

2.1 Between-subjects analysis of Learning and Attitudes

Table 1 shows the results for pre- and post-test scores for the three groups, i) Intervention Group (used the tutor with interventions every six problems), ii) Tutor Control Group (used the tutor w without the interventions) and iii) No-Tutor Control (matched students who used no software).

Group	Math Pretest	Math Posttest	MCAS Passing Rate
No Tutor Control			76% (N=38)
Tutor Control	40% (20) (N=40)	40% (28)* (N=40)	79% (N=34)
Tutor Intervention	33% (19) (N=36)	42% (22)* (N=36)	92% (N=24)

Table 1. Means and standard deviations in performance measures before and after tutoring

The overall learning gain (Posttest-Pretest) for the experimental group was 7%, while the Tutor Control group showed no improvement (Table 1). These learning gains are smaller than what we had observed previous years (15% in about the same amount of time). We think that the overall posttest scores underestimate students' ability because: 1) it was online, and we observed gaming behavior particularly in the posttest; 2) it was started at the end of a period, when students were already tired. The fixed sequencing might have affected learning gains also (in contrast to the adaptive sequencing of problems). In any case, the low posttest scores do not prevent us from carrying out a between-subjects comparison. An ANCOVA was used to analyze the difference between the learning gains between the two groups (tutor Intervention and tutor-control). The dependent variable was posttest score (percent correct), with group as an independent variable, and pretest as a covariate. The test of between subjects indicated a significant difference in posttest score between the tutor-control and tutor-Intervention groups ($F=4.23$, $p=.04$), suggesting that there is a significant difference in learning gains favoring the interventions-enhanced tutor group.

Because the experiment was carried out days before a statewide-standardized test exam (MCAS), we collected standardized scores as well for students who took the exam, including the matched group of students (same level, same teachers) who did not use the tutor. Note that students in the Interventions group had a higher average learning gain (7% more than the control group) and higher passing rate at the MCAS exam (92% vs. 79%) than their counterparts in the control group, and higher than the no tutor control group (92% vs. 76%). A Chi-square test was carried out on the cross-tabulation between MCAS passing (1 for passed, 0 for did not pass) and version (Tutor Intervention and No Tutor Control), indicating a marginally significant difference ($p=.12$) that suggests that the Tutor Intervention group had significantly higher passing rate.

Table 2 shows results of the surveys that were higher for the Interventions group. Students in the Interventions group agreed more with the statements such as "the Wayang tutor was smart and friendly". They also had significantly higher learning orientation scores in two items that measure performance vs. learning orientation (Mueller&Dweck, 1998). Marginally significant differences were observed for students thinking they have learned with the Wayang tutor and beliefs about the helpfulness of the help, all favoring the Interventions group. No significant differences were encountered for questions about 'computers caring about myself', 'Wayang is genuinely concerned about my learning', feeling of control over computers, mathematics liking, 'the tutor is concerned about my learning', self-concept about mathematics ability, or self-efficacy. These last ones are deeper feelings about oneself, and the Interventions don't seem to have impacted at that level, but at a simpler level of perceptions of the system, its helpfulness, and their willingness to learn.

Survey question item	Tutor Interventions	Tutor Control
"The Wayang Tutor is friendly" ANOVA: F=6.5, p=.01**	4.8 (1.0) N=21	3.9 (1.4) N=35
"The Wayang tutor is smart" ANOVA: F=6.5, p=.01**	5.1 (1.0) N=21	4.3 (1.3) N=35
Learning Orientation (average over 2 items) ANOVA: F=4.2, p=.045*	0.60 (.8) N=21	0.39 (.6) N=37
Did you learn how to tackle math problems by using the Wayang system? ANOVA: F=2.9, p=.09 (marginal)	3.5 (.74) N=22	3.1 (.82) N=37
Helpfulness of the help (average over 3 items) ANOVA: F=2.5, p=.1 (marginal)	4.2 (.73) N=21	3.8 (.9) N=36

Table 2. Means and standard deviations for responses to post-tutor surveys

2.2 Within-subjects analysis of changes in engagement

The second analysis has to do with students' variation of engagement within the tutor. If the interventions were effective, it would be expected that there is a difference in indicators of engagement, from the problem before the intervention was given to the problem after the intervention was given. Some other authors have suggested time as an indicator of engagement (Beck, 2004). If the student is not engaged, then the student will either skip fast through the problem, or abuse help quickly to get to the correct answer, and all gaming behaviours are related to timing issues (Baker, 2006). Spending long time in a problem is an insufficient condition to learning (the student might be off-task), but it is also a necessary condition (if the student doesn't invest time on the problem, they will surely not learn). Students might be off task when spending much time in a problem, but if outlier cases of time spent per problem are discarded, or median/quartile statistics are taken into account instead of means/standard-deviations, we can look at the time spent per problem variable with some confidence that what we are looking at is engagement and focus on the problem.

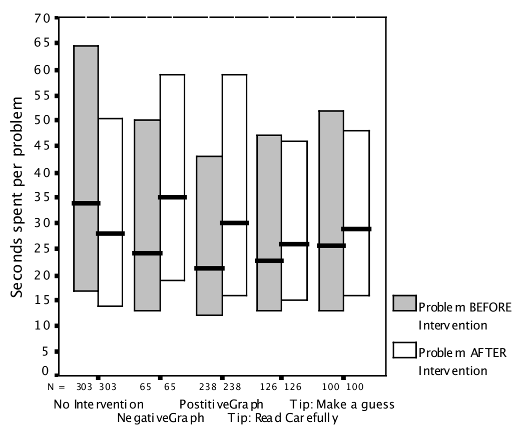


Figure 2. Medians and Quartiles for time spent in the problem immediately before and after the intervention

2.2.1 Time spent per problem. Difference in subsequent problems.

The main question is how a student's behavior changes during the time spent per problem from before to after the intervention. How do different interventions affect the time spent in a problem? What is the difference between the times spent during the two problems? The first two boxes in the Box Plot of Figure 2 show the median and quartile seconds spent per problem for 303 random pairs of subsequent problems, *for students in the tutor-control* group. These two boxes suggest that students tend to get more disengaged as the session progresses (median and quartiles are lower in the second problem), by a median 6 seconds less in the next problem. The eight boxes to the right correspond to the seconds spent in 469 problems immediately before and immediately after each

intervention, for students in the Interventions group. Clearly, there is a reverse effect of students decreasing time in subsequent problems: students increase the median time spent in the problem after seeing any intervention.

There are two reasons why carrying out regular statistical analyses is difficult with this kind of data. First, a standard analysis of variance or means comparison relies on the fact that probability distributions should be normal. This is not the case of this data set, as distributions for time-based variables have minimums at zero and long tails for outlier times. Second, each case corresponding to the time spent in a pair of problems is not independent from each other, because there are several pairs corresponding to each student. Thus, ANOVA statistical analyses are merely exploratory, but we believe worth as an indication of the strength of the results.
Taking this into account, and after removing outlier time values, repeated measures ANOVA indicated a significant difference in time change within ($F=8.79$, $p=.003$) and between the two groups ($F=7.3$, $p=0.007$). However, note that the shift in time is more pronounced for the graph interventions than for the tip interventions. A paired-samples t-test gave a significant difference for time spent from the problem before to the problem after a Graph Intervention ($t=-2.9$, $p=0.004$), but not for before and after the tips ($t=.69$, $p=.49$).

2.2.2 Do disengaged students game less after seeing an intervention?

If time spent in a problem reflects level of engagement, students apparently become more disengaged as time progresses in the tutoring session if the tutor does nothing except presenting additional problems. The main question is whether there is a way to show that students reduce specific disengagement actions. In Wayang, we may answer this question by referring to a model developed by Johns and Woolf (2005) from past data of Wayang users, which allows to classify students in either an 'engaged' state or otherwise three types of disengagement: i) quick-guess, ii)hint-abuse, and iii) skipped-problem. A hidden markov model infers the probability that a student is in any of these states for each problem, depending on their behavior on the current problem and the motivation state in the previous time step. The student is classified as engaged/disengaged based on the engagement state of highest likelihood. Disengagement states are similar to those identified by Baker (2006) regarding gaming the system –all measures of disengagement that correlate to learning.

An important question to answer regards the potential of these interventions to re-engage students compared to the control group –what are the odds that a student will turn from one disengagement state back to the engaged state after seeing an intervention, compared to those students who did not see any interventions at all. 347 episodes of 6 subsequent engagement states were extracted for the control group (corresponding to six subsequent problems), and 433 sequences were extracted for the experimental group. The latter sequences had the peculiarity that they corresponded to three problems before an intervention was shown, and three problems after an intervention was shown. Figure 3 shows how many students were classified as engaged or disengaged at each time step, for each group. Note that the probability that a student will continue to stay in the same engagement state or switch to the other state (labelled arrows in the figure) were computed from students' data, for example: 79 students from the interventions group were considered disengaged in time t_4. 49% of those 79 students switched to being engaged in time t_5. Thus, the probability of re-engaging at t_5, or $P(E_{t5}|\sim E_{t4})$, is 0.49. The other transition probabilities are the probability of disengaging, (e.g. $P(\sim E_{t5}|E_{t4})$), staying engaged (e.g. $P(E_{t5}|E_{t4})$), staying disengaged (e.g. $P(\sim E_{t5}|\sim E_{t4})$) and were computed in a similar way. All probabilities were rounded to the second decimal.

	t_4	T_5	t_6		t_7	t_8	t_9
Interventions	0.18	0.18	0.20		0.18	0.21	0.20
Control	0.17	0.19	0.22		0.22	0.19	0.18
Difference	*0.01*	*-0.01*	*-0.02*		*-0.04*	*0.02*	*0.02*

Table 3. Probability of a student being disengaged at t_4, …, t_9

Arrows going from bottom states to top states in Figure 3 indicate the odds that a student will become re-engaged in the task, or $P(E_{ti}|\sim E_{ti-1})$. Note that these re-engagement probabilities are generally larger for the interventions group, except for the transition between t_5 ad t_6, where they are equal. Noticeably, the largest difference between the two groups happens immediately after the intervention, between t_6 and t_7 (there is a 16% advantage for the interventions group in t_6-t_7, 5% advantage in t_7-t_8, 11% in t_8-t_9). However, note that in general students in the interventions group are also more likely to disengage by a small margin. This suggests that once students are engaged, that engagement status might be hard to maintain. The interventions group is 1% more likely to disengage in t_6-t_7 than the control group, 3% in t_7-t_8, 4% in t_8-t_9. It is possible that the interventions may have only a short-term positive impact on engagement, though it is also possible that exposing students who are already engaged to an intervention might be unproductive. These two reasons help explain why there is not much of a difference between the groups in total percentage of engaged students at each time step (Table 3), nor an 'overall' increase in total students considered engaged for the interventions group from time t_4 to time t_9. Students in the

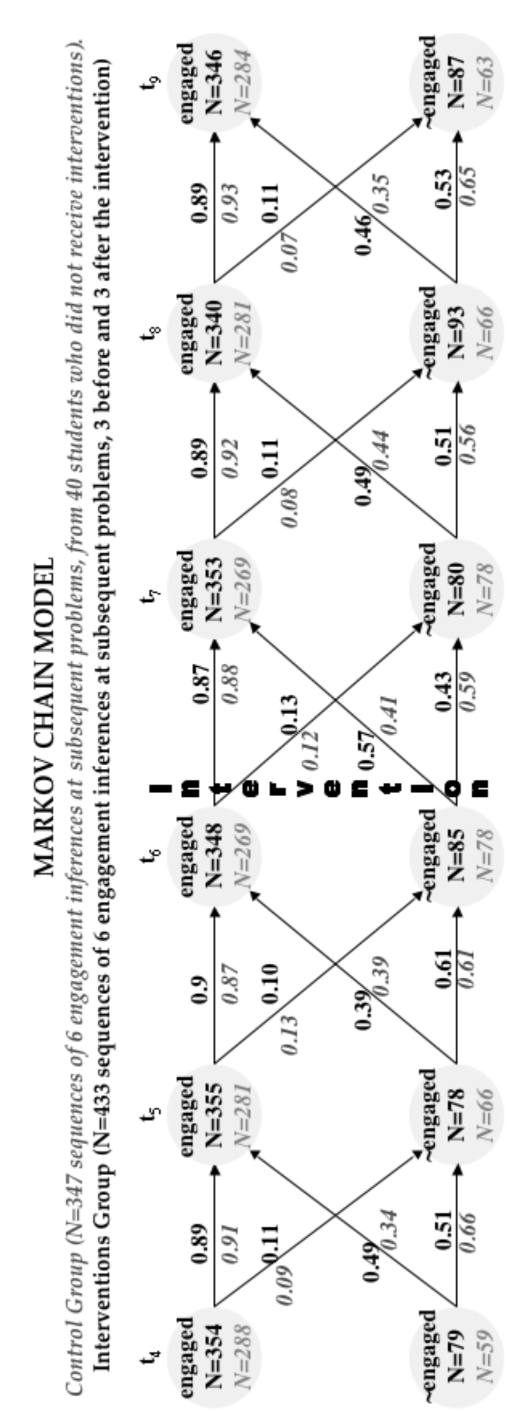


Figure 3. probabilities of disengagement, re-engagement, or staying in the same state (computed from data)

experimental group clearly oscillated more between engaged and disengaged states, as both probabilities of disengagement and re-engagement are generally higher than the control group. There is much more movement between engaged and disengaged states for students in the experimental group.

Another interesting question to explore is whether students modified specific kinds of disengagement behaviors. With that thought in mind, Table 4 shows transition probabilities between the before-problem and the problem after the intervention --between t_6 and t_7. These transition probabilities were computed discriminating by type of disengagement. For instance, the first column in table 4 indicates the probability of transitioning from any state into the engaged state after seeing an intervention. At the same time, the first column in Table 5 is our control, the probability of transitioning into an engaged state, after not seeing any interventions).

Table 6 is the difference between the previous matrices: positive values indicate that the probability is higher for the motivational group, negative values indicate the opposite. Therefore, the first column in Table 6 shows that students re-engage more (transition more from disengaged states into the engaged state) after seeing an intervention. Students who quick-guessed in the problem before the intervention had a 12% higher chance to re-engage than those in the control group (same problem, but no intervention). Students who abused help in the problem before the intervention had 10% higher chance of re-engaging in the following time step than the control group. Students who skipped the problem before the intervention have 51% higher chance of being considered engaged in the following time step than the control (though it is a low number of students skipping problems, 5 out of 7 skipping-students re-engaged after the intervention vs. 1 out of 5 in the control group). Also, as the second column in Table 6 shows, students have less chance of transitioning into the quick-guess state after an intervention than in the control group. Table 6 also shows that students in the Interventions group are less prone to stay in the same

disengagement state (bold diagonal). This may mean transitioning from one form of disengagement state into another form of disengagement. For instance, students have 10% higher chance to transition from hint-abuse to skipping the problem after seeing an intervention. However, this might just be due to the low number of cases, as students did not abuse help much. Fortunately, most students who skip the problem before the intervention (71%) re-engage in the following problem (51% higher than the control group).

		Problem after the Intervention was seen			
		Engaged	Quick-Guess	Hint Abuse	Skipped
Problem Before Intervention	Engaged (348)	**.88 (305)**	.06 (22)	.04 (13)	.02 (8)
	Quick-guess (58)	.53 (31)	**.45 (26)**	0 (0)	.02 (1)
	Hint Abuse (20)	.60 (12)	.10 (2)	**.20 (4)**	.10 (2)
	Skipped (7)	.71 (5)	0 (0)	0 (0)	**.29 (2)**
	Totals (433)	*.82 (353)*	*.12 (50)*	*.10 (17)*	*.03 (13)*

Table 4. Transition Probabilities between t_6 and t_7 (before and after Intervention), Interventions group.

		Problem after the Intervention was seen			
		Engaged	Quick-Guess	Hint Abuse	Skipped
Problem Before Intrvention	Engaged (269)	**.88 (237)**	.09 (23)	.03 (8)	.00 (1)
	Quick-guess (61)	.41 (25)	**.57 (35)**	0 (0)	.02 (1)
	Hint Abuse (12)	.50 (6)	.25 (3)	**.25 (3)**	0 (0)
	Skipped (5)	.20 (1)	.20 (1)	0	**.60 (3)**
	Totals (347)	*.78 (269)*	*.18 (62)*	*.03 (11)*	*.01(5)*

Table 5. Transition Probabilities between t_6 and t_7 in the Control Group

		Problem after the Intervention was seen			
		Engaged	Quick-Guess	Hint Abuse	Skipped
Problem Before Intervent ion	Engaged	**0.00**	-0.02	0.01	0.02
	Quick-guess	0.12	**-0.13**	0.00	0.00
	Hint Abuse	0.10	-0.15	**-0.05**	0.10
	Skipped	0.51	-0.20	0.00	**-0.31**
	Totals	*0.04*	*-0.06***	*0.01*	*0.02*

Table 6. Matrix Difference: Intervention – Control transition probabilities (t-test p<.01 **)

Because these transition probabilities can also be considered means computed over sample data, an independent samples t-test can help us to explore if these differences are significant. Again, there is an assumption that the cases are independent when they really are not (several cases correspond to the same student) which is why this is an exploratory analysis. Only about 10% of students were classified as disengaged at any time step, so significant differences are rare due to low number of cases. Still, a t-test revealed that students quick-guessed significantly less in the problem after the intervention, 6% less than in the matched problem in the control group. Table 7 breaks down the number of quick-guess cases in the problem after (how many quick-guesses happened after a tip intervention, and how many after a graph intervention?). There is a significant difference in mean percent of quick-guesses after a graph intervention (Graphs – Control= -.08, p=0.005), but not after the tip-interventions (Tips – Control = -.04, p=0.18). It is the problems after a student received a graph performance-monitoring intervention, not the problems after receiving a tip intervention, in which students have lowest amount of quick guesses (compared to the engagement state in matched problems after no intervention).

Kind of Intervention	N	Mean % of quick-guess	Std. Deviation	ANOVA
No Intervention-Control	347	.18	.38	F=6.3 p=.02*
Tip Intervention	191	.14	.31	
Graphs Intervention	242	.10	.30	

Table 7. Means, std. deviations and univariate ANOVA for mean quick guesses in t_7
(* significant difference at p<0.05)

3. Discussion and Conclusion

Despite the fact that students in the Interventions group started off with slightly lower pre-test scores, they achieved higher posttest scores than the control group, higher passing rates in standardized tests, higher learning-orientation in a post-tutor survey, had the feeling the tutor was more helpful and that they learned more, and attributed more human-like characteristics to the tutoring software. We attribute these differences to the non-invasive Interventions provided to students in between problems. Students have a higher chance to become re-engaged immediately after seeing an intervention, regardless of whether they quick-guessed, abused help or skipped the problem before the intervention. It is in particular the progress-monitoring interventions that helped students have less quick-guesses, and higher changes in time spent per problem.

Showing students their progress with a simple open-learner model can be considered a type of meta-cognitive feedback. It encourages self-monitoring by highlighting that both the software and themselves can keep track of whether they are answering correctly or incorrectly, and whether they are making progress. Our simple progress charts talk to the student in terms of correctly answered questions so that students can clearly link their actions to the chart, and the chart to their learning progress. We think it is showing students' performance in particular that re-engages students because being hinted of ones' progress together with tips provided for lower quick-guesses and higher improvements in time spent per problem than the tips by themselves. Time spent per problem especially improved if the graph provided was negative --indicated that no improvement had been made. One possible explanation is that negative feedback may have encouraged students to be more cautious, to reduce the chances of getting negative feedback again. However, it is important to note that the data analyzed indicates no clear benefit in sustaining engagement. A measure of success for future interventions will be not only how much it can re-engage a student in the following problem, but for how long the student continues to be in such state. One other important aspect is that the timing of interventions may be relevant. They may be more beneficial to show immediately after the problem where the student is disengaged, though this has not been established yet. The decision of whether to present an intervention after each problem or not, depending on the assessed disengagement state and other factors may be relevant to enhance engagement, and learning and motivation in the long run.

In the near term we plan to design of new kinds of interventions, as well as analyzing which ones are beneficial for which students, and at which moments. Firstly, something to revisit is the accuracy of the graph interventions in assessing students' knowledge. Our current graph interventions are very simple, and possibly inaccurate some times at showing the student how much they have learned --comparing percentage correct responses in the last problems is not an accurate enough measure to assess how much students know, in the face of large variation of problem difficulties and skills involved. On the other hand, we believe that merely reporting number of correct responses offers a very tangible measure for students to understand and relate to their actions and their progress, so we want to find a representation to externalize students' progress that is thorough but also easily understandable. Secondly, new kinds of meta-cognitive feedback (beyond self-monitoring) could be beneficial for students, such as specific feedback on planning how to solve a specific kind of problem, or motivational feedback that acknowledges students' feelings of frustration, or helps establish productive attributions for failure and success (Mueller and Dweck, 1998).

We conclude that non-invasive interventions that help students reflect on their progress are beneficial to students learning, attitudes towards learning, and towards the tutoring software. These research results provide valuable insight in support of in-between problem meta-cognitive feedback, and in support of open learner models for the design of effective software interactive learning environments.

References

Arroyo, I., Beal, C. R., Murray, T., Walles, R., Woolf, B. P. (2004). Web-Based Intelligent Multimedia Tutoring for High Stakes Achievement Tests. In Proceedings of the 7th International Conference on Intelligent Tutoring Systems, pages 468-477.

Baker, R.; Corbett, A.; Koedinger, K. (2004) Detecting student misuse of intelligent tutoring systems. In Proceedings of the 7th International Conference on Intelligent Tutoring Systems, pages 43--76.

Baker, D.J., Beck, J. (2006) Adapting to When Students Game an Intelligent Tutoring System. Proceedings of the 8th International Conference on Intelligent Tutoring Systems, 392-401

Beck, J. (2004) Using response times to model student disengagement. Proceedings of the ITS2004 Workshop on Social and Emotional Intelligence in Learning Environments, August, 2004

de Jong, & van Joolingen, W. R. (1998). Scientific Discovery Learning with Computer Simulations of Conceptual Domains . Review of Educational Research, 68, 179-201.

Johns, J.; Woolf, B.P. (2006). A Dynamic Mixture Model to Detect Student Motivation and Proficiency. Proceedings of the Twenty-first National Conference on Artificial Intelligence (AAAI-06), Boston, MA.

Hartman, H. J. (2001). Metacognition in Learning and Instruction, Hartman, HJ (ed). Springer.

Mueller, C.M., Dweck, C.S. (1998). Praise for intelligence can undermine children's and performance. Journal of Personality and Social Psychology , 75 (1), 33-52.

Roll, I., Aleven, V., McLaren, B. M., Ryu, E., Baker, R., and Koedinger, K. R. (2006). The Help Tutor: Does Metacognitive Feedback Improve Students' Help-Seeking Actions, Skills and Learning?

Wigfield, A. & Karpathian, M. (1991). Who am I and what can I do? Children's self-concepts and motivation in academic situations. Educational Psychologist, 26, 233--262.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Can Help Seeking Be Tutored? Searching for the Secret Sauce of Metacognitive Tutoring

Ido ROLL, Vincent ALEVEN, Bruce M. MCLAREN, Kenneth R. KOEDINGER
Human-Computer Interaction Institute, Carnegie Mellon University

Abstract. In our on-going endeavor to teach students better help-seeking skills we designed a three-pronged Help-Seeking Support Environment that includes (a) classroom instruction (b) a Self-Assessment Tutor, to help students evaluate their own need for help, and (c) an updated version of the Help Tutor, which provides feedback with respect to students' help-seeking behavior, as they solve problems with the help of an ITS. In doing so, we attempt to offer a comprehensive help-seeking suite to support the *knowledge*, *skills*, and *dispositions* students need in order to become more effective help seekers. In a classroom evaluation, we found that the Help-Seeking Support Environment was successful in improving students' declarative help-seeking knowledge, but did not improve students' learning at the domain level or their help-seeking behavior in a paper-and-pencil environment. We raise a number of hypotheses in an attempt to explain these results. We question the current focus of metacognitive tutoring, and suggest ways to reexamine the role of help facilities and of metacognitive tutoring within ITSs.

Keywords. Help Seeking; Self-Assessment; Metacognition; Self-Regulated Learning; Intelligent Tutoring Systems; Cognitive Tutors; Empirical Study

Introduction

One of the challenges students face while working with an Intelligent Tutoring System (ITS), as part of regulating their learning, is choosing what actions to perform: using an online help resource, approaching a teacher or peer, or attempting to solve the next problem step. Seeking the right form of help at the right time is known to be associated with better learning [1; 17]. However, students' help-seeking behavior is far from ideal [3]. Within an ITS, students often ask for over-elaborated help when none or little is needed, but avoid asking for necessary help. This maladaptive help-seeking behavior appears to be consistent across domains and students [12].

One approach to help students improve their help-seeking behavior is to design the system so that it limits their opportunities for maladaptive help seeking. For example, Wood [17] created a "contingent tutor" that adapts the hint level to the student's proficiency, and the Cognitive Tutor [8] has a built-in two seconds delay that prevents repeated fast hint requests. However, this approach does not necessarily lead to an improvement in students' help-seeking skills. A different approach separates the cognitive and the metacognitive demands of the task. For example, Reif and Scott [10] and Gama [7] separate the practice opportunities of monitoring or planning skills from

those of domain skills. However, it is important that the help-seeking behavior is practiced within the learning context.

In this project we attempt to help students acquire and spontaneously apply better help-seeking skills that persist beyond the scope of the tutoring system itself. We do so in the context of the Geometry Cognitive Tutor, which, like other Cognitive Tutors [8], gives students on time, tailored feedback on their problem-solving actions. It does so by tracing students' actions relative to a cognitive model. The Geometry Cognitive Tutor has two help mechanisms: (i) several levels of contextual hints are available for students, with the most elaborated one giving away the answer, and (ii) the glossary is a searchable geometry knowledge base, much like an online dictionary.

Following the Cognitive Tutor pedagogy [8], our first attempt was to teach help seeking by giving students feedback on their help-seeking behavior (specifically, their help-seeking errors). The Help Tutor [11], a Cognitive Tutor in its own right, identifies recommended types of actions by tracing students' interaction with the Geometry Cognitive Tutor in real time relative to a metacognitive help-seeking model [1]. When students perform actions that deviate from the recommended ones, the Help Tutor presents a message that stresses the recommended action to be taken. Messages from the metacognitive Help Tutor and the domain-level Cognitive Tutor are coordinated, so that the student receives only the most helpful message at each point [2].

Previous evaluations of the Help Tutor led to mixed results. On the one hand, the Help Tutor did as planned – gave feedback on learning events that were associated with poorer learning outcomes, and led to a reduction in help-seeking errors. On the other hand, working with the Help Tutor did not lead to improved learning of domain knowledge, nor to better declarative knowledge of the ideal help-seeking behavior [11].

1. The Help-Seeking Support Environment

These results suggest that a more comprehensive approach is required in order to help students become better help seekers, one that includes substantial declarative and dispositional components, in addition to a procedural one. To address these needs we designed the Help-Seeking Support Environment (HSSE), which includes the following components:

The Updated Help Tutor stresses help-seeking principles and benefits, and not merely the error and the recommended action. For example, the message ""Slow down, slow down. No need to rush" was changed to: "It may not seem like a big deal, but hurrying through these steps may lead to later errors. Try to slow down."

The Self-Assessment Tutor. As shown by Tobias and Everson [15], the ability to correctly self-assess one's own ability is correlated with strategic use of help. While

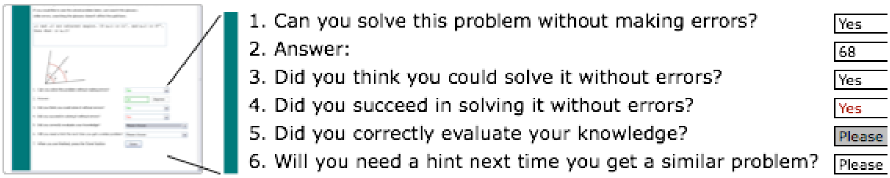


Figure 1. The Self Assessment tutor scaffolds self-assessment in four stages: *Prediction*, in which the student predicts whether she knows how to solve the problem (Q. 1); *Attempt*, in which the student attempts to solve the problem (Q. 2); *reflection*, in which the student contrasts her actual performance with her prediction (Q. 3-5); and *projection*, in which the student projects from the existing experience on future need for help when encountering similar problems (Q. 6).

working with the Self-Assessment Tutor, students are asked to assess their ability on the target set of skills, compare this assessment to their actual performance, and make appropriate choices regarding their subsequent learning process ([13], see Figure 1).

Declarative instruction. As White and Frederiksen demonstrated [16], reflecting in the classroom environment on the desired metacognitive process helps students internalize it. With that goal in mind, we created a short classroom lesson about help seeking with the following objectives: to give students a better declarative understanding of desired and effective help-seeking behavior (e.g., “take the time to think before you act”); to improve their dispositions and attitudes towards seeking help (e.g., “Struggling is part of the learning process. You will not learn by guessing or abusing hints, even if you get the answer right”); and to frame the help-seeking knowledge as an important learning goal, alongside knowledge of geometry. The instruction comprises a 4 minutes video presentation with examples of productive and faulty help-seeking behavior and the main help-seeking principles, followed by 5-10 minutes of teacher-led discussion.

The HSSE curriculum begins with the declarative instruction, followed by interleaved Self Assessment and Cognitive Tutor + Help Tutor sessions, with the Self Assessment sessions taking about 10% of the students’ time (Table 1). In order to help students notice the broad relevance and domain-independent nature of the help-seeking knowledge, we made the HSSE available across two instructional units.

Table 1: The study procedure. ⓘ - Declarative instruction; ✍ - Self-assessment preparatory session. Table is not to scale, i.e., self-assessment and instructional activities took about 10% of the class time

Week:		1	2	3	4		5-9	10	11	12	13	
		Unit 1 (Angles)						Unit 2 (Quadrilaterals)				
Help group	Pretest 1	ⓘ	✍	✍	✍	Posttest 1, Pretest 2	Break	ⓘ	✍	✍	✍	Posttest 2
		Cognitive Tutor + HSSE						Cognitive Tutor + HSSE				
Control group		Cognitive Tutor						Cognitive Tutor				

2. Method

Participants. The HSSE was evaluated with 67 students from 4 classrooms instructed by 2 teachers, from a rural vocational school. All students, 10th and 11th graders, were enrolled in the Cognitive Tutor Geometry class, and thus were familiar with the Cognitive Tutor and its interface.

Design. Since the HSSE includes teacher-led classroom instruction that is not given to the Control condition, the study was done in a between-class fashion. Two classes (with 29 students) were assigned to the Help condition (Cognitive Tutor + HSSE) with the remaining two classes (38 students) were assigned to the Control condition.[1] Classes were assigned in consultation with the teachers, attempting to

[1] An additional class was assigned to the Help condition. We discarded its data due to lack of effort according to the teacher (who reported that the students did not make a serious effort to do well on the tests) and the data. For example, students in this class left blank as many as 38% of the problems on posttest1, compared with only 11% in the other classes. This is probably due to lack of effort, since most of the items left blank were concentrated at the end of the form, regardless of the difficulty level of the counter-balanced problems. Including the class in the analysis does not affect the results qualitatively.

control for number of students and time of day. One class that was reported by the teachers to be of lower ability was assigned to the Help condition. Each teacher taught classes in both conditions.

Procedure. The study spanned a period of three months and two instructional units, during which students in the Help condition used the HSSE for about 15 academic hours (see table 1). Each instructional unit took about a month, with a month in between during which students prepared for the standardized state test. The declarative instruction on help seeking was given twice during the study period in the Help condition classes, at the beginning of each unit. The total time spent on the study was kept constant across classes and conditions.

Assessment design. Students' geometry knowledge was assessed three times across the study: Prior to unit 1 (pretest on unit 1), in between the two units (posttest on unit 1 and a partial pretest on unit 2), and following unit 2 (posttest on unit 2). The tests included "regular" geometry problem-solving items as well as a number of deep-understanding measures: (i) *reason items*. Students were asked to state the theorem they used (ii) *data insufficiency items*. Students were told (on all problems) that if there is not enough information to answer the problem they should state so by writing 'No'. About 10% of all steps in the tests lacked sufficient information. (iii) *conceptual understanding items*. Unit 2 included items on which students needed to demonstrate conceptual understanding, for example, by matching up diagrams and geometric properties with given geometric shapes.

Students' help-seeking behavior was evaluated using embedded hints in the test [11]. Several test items appeared in three hint conditions, counterbalanced between forms: conventional *No hint* items; *Request hint* items, which contained a hint that was covered by a sticker (students were told that removing the sticker would cost them 10% of the item's score); and *Free* and open hint items (see Figure 2).

Last, students' declarative help-seeking knowledge was evaluated using hypothetical help-seeking dilemmas. These multiple-choice questions asked students to choose the appropriate action to perform in response to situations such as repeated errors, easy steps, verbose hints, etc. These situations were not discussed during the instruction.

The tests were piloted for difficulty level and comprehensibility with students from a parallel class at the same school.

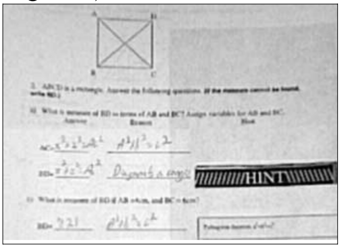


Figure 2. A typical test item with three hint conditions: No hint, Request hint (students were told that removing the sticker will lead to a 10% reduction of their grade on that step), and Free hint. Hint conditions were counterbalanced between test forms.

3. Results

In the study we addressed the following questions: (a) Did the HSSE affect learning at the domain level? (b) Did it affect help-seeking behavior as measured by the use of embedded hints in the paper test? And (c) did it affect declarative knowledge of good help-seeking? Due to absences, 53, 52 and 43 forms (of the 67 registered students) were collected for the three respective tests, divided about evenly between conditions.

Learning gains. As detailed above, the test included common geometry problem-solving items and robust-learning items in the form of Reason, Data Insufficiency items, and in unit 2, also Conceptual Understanding items. As seen in Figure 3, both

groups improved significantly from pretest (time 0) to posttest1 (time 1) on the Problem solving and Reason items (Problem solving: $t(45)=3.1$, $p=0.004$; Reason: $t(45)=5.6$, $p<0.0005$). The scores at posttest2 (time 2) cannot be compared to those at posttest1, since they pertain to a different instructional unit. The only item types at time 2 to have a pretest at time 1 were the Conceptual items, in which significant learning is observed ($t(34)=6.8$, $p<0.0005$). No learning was observed on Data Insufficiency items.

The HSSE had no effect on any of the scores. There were no differences in learning between the groups, with the Help group scoring a bit lower on the procedural problem-solving items on all tests, as was expected, given that one of the classes in this condition included many lower-achieving students. There was also no significant interaction that includes condition.

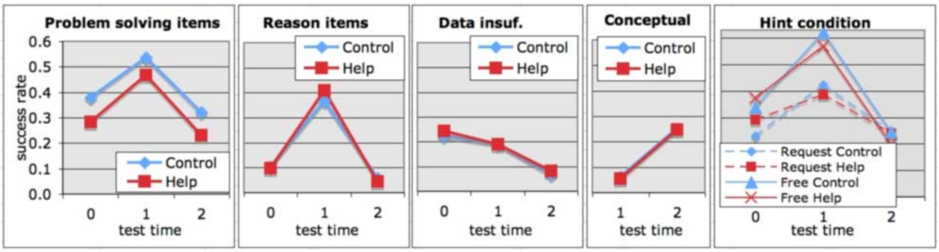


Figure 3. Learning gains on the different measures. From left to right: Problem solving items, Reason items, Data Insufficiency items, Conceptual items, and items with embedded hints. Posttest 2 evaluated a different unit form posttest 1, and thus used a different test. While there is significant learning from pre to post, there are virtually no differences between conditions that are not accounted for by pretest differences.

Hint behavior. A small number of test items had a hint manipulation, counter-balanced between forms: No hint, Request, and Free (see Figure 2.) To control for domain-level difficulty, we measured the added value of having a hint by computing the ratio between scores on items with a hint (either Request or Free) and items that did not have a hint. This measure had been validated earlier by showing that it correlates with online help-seeking behavior in the tutor [11]. As before, no differences were observed between conditions. Also, no interaction that includes condition reached significance.

Figure 3 shows scores on items with hints. Overall, 12 measures of hint scores were used (2 hint types * 2 conditions * 3 test-times). Scores on items with hints were lower than on the same items with no hints on 7 out of the 12 hint measures. In other words, interestingly, having some form of a hint hindered performance on more than half of the measures in the paper test. This is probably due to the novelty of having hints in the tests, although this was not the case in previous uses of this method.

Declarative Knowledge of Help Seeking. The only significant difference between the groups was found in the Declarative Help-Seeking knowledge assessment, evaluated by means of the hypothetical help-seeking dilemmas questionnaire. The Help group students scored 74% whereas the Control group students scored only 47% on that test (Figure 4; $F(1,31)=6.5$, $p<0.02$)[2]. Thus, students in the Help

Figure 4. Significant differences in help seeking declarative assessment show that the Help group gained a better understanding of the help-seeking process.

[2] Not all students filled in the questionnaire. When including all students in the analysis, including those who skipped it, the effect still holds, although is no longer significant (Help: 55%; Control: 37%; p=0.14).

condition demonstrated better understanding of the help-seeking process and not merely greater inclination to use hints.

4. Conclusions

This paper discusses a comprehensive metacognitive environment to support help-seeking behavior, and the outcomes of a classroom evaluation done with the environment. The environment supports and links three types of help-seeking knowledge: declarative, procedural, and dispositional. It combines three instructional approaches: classroom discussion, preparatory self-assessment sessions, and feedback on help-seeking errors in the main tutoring environment (the Geometry Cognitive Tutor). A classroom evaluation with 67 students revealed that use of the HSSE contributed only to students' declarative help-seeking knowledge. We found no differences between the groups with respect to their domain-specific learning or their help-seeking behavior on the paper-and-pencil test.

These somewhat disappointing results raise an important question: Why did the environment not lead to an improvement in learning and in help-seeking behavior in the paper-test measures? One possible explanation may be that the HSSE imposes excessive cognitive load during problem solving. Clearly, the learning process with the HSSE is more demanding compared to that with the conventional Cognitive Tutor alone, since more needs to be learned. However, much of the extra content is introduced during the classroom discussion and self-assessment sessions. The only extra content presented during the problem-solving sessions are the Help Tutor's error messages, but they are not expected to increase the load much, especially given that a prioritization algorithm makes sure students receive only one message at a time (either from the Help Tutor or the Cognitive Tutor).

The HSSE is likely to have improved students' help-seeking behavior while working with it. Although we have not yet analyzed the log files of this study, in a previous study, the Help Tutor by itself (without the additional components of the HSSE) was shown to have an effect on online behavior [11]. Also, the Help Tutor makes it harder to commit certain types of help-seeking errors (such as immediate repeated hint requests) due to its feedback. It is therefore reasonable to assume that students in the Help group committed fewer help-seeking errors than Control group students. But if that is so, why then did the improved online help-seeking behavior not lead to improved learning gains? Several hypotheses can be put forward in this regard, forming two lines of reasoning: *The role of help seeking in ITS*, and *the focus of metacognitive tutoring in ITS*.

The role of help seeking in ITS. Hints in tutoring systems have two objectives: to promote learning of challenging skills, and to help students move forward within the curriculum (i.e., to prevent them from getting stuck). While the latter is achieved easily with both the Cognitive Tutor and the HSSE, achieving the first is much harder. Surprising as it may sound, it is not yet clear what makes a good hint, and how to sequence hints in an effective way. It is possible that the hints as implemented in the units of the Cognitive Tutor we used are not optimal. For example, there may be too many levels of hints, with each level adding too little information to the previous one. Also, perhaps the detailed explanations are too demanding with regard to students' reading comprehension ability. It is quite possible that these hints, regardless of how they are being used, do not contribute much to learning. Support for that idea comes

from Schworm and Renkl [14], who found that explanations offered by the system impaired learning when self-explanation was required. The Geometry Cognitive Tutor prompts for self-explanation in certain units. Perhaps elaborated hints are redundant, or even damaging, when self-explanation is required.

It is possible also that metacognitive behavior that we currently view as faulty may actually be useful and desirable, in specific contexts for specific students. For example, perhaps a student who does not know the material should be allowed to view the bottom-out hint *immediately*, in order to turn the problem into a solved example. Support for that idea can be found in a study by Yudelson et al. [18], in which medical students in a leading med school successfully learned by repeatedly asking for more elaborated hints. Such "clicking-though hints" behavior would be considered faulty by the HSSE. However, Yudelson's population of students is known to have good metacognitive skills (without them it is unlikely they would have reached their current position). Further evidence can be found in Baker et al. [4], who showed that some students who "game the system" (i.e., click through hints or guess repeatedly) learn just as much as students who do not game. It may be the case that certain gaming behaviors are adaptive, and not irrational. Students who use these strategies will insist on viewing the bottom-out hint and will ignore all intermediate hints, whether domain-level or metacognitive. Once intermediate hints are ignored, better help-seeking behavior according to the HSSE should have no effect whatsoever on domain knowledge, as indeed was seen. It is possible that we are overestimating students' ability to learn from hints. Our first recommendation is to re-evaluate the role of hints in ITS using complementary methodologies such as log-file analysis (e.g., dynamic Bayes nets [6]); tracing individual students, experiments evaluating the effect of different types of hints (for example, proactive vs. on demand), and analysis of human tutors who aid students while working with ITS.

The focus of metacognitive tutoring in ITS. Students' tendency to skip hints suggests that perhaps the main issue is not lack of knowledge, but lack of motivation. For students who ignore intermediate hints, metacognitive messages offer little incentive. While the HSSE can increase the probability that a proper hint level appears on the screen, it has no influence on whether it is being read. Students may ignore the messages for several reasons. For example, they may habitually click through hints, and may resent the changes that the HSSE imposes. This idea is consistent with the teachers' observation that the students were not fond of the HSSE error messages. The test data discussed above provides support for this idea. On 7 out of the 12 hint evaluations students scored lower on items with hints than on items with no hints. A cognitive load explanation does not account for this difference, since the Request hints did not add much load. A more likely explanation is that students chose to skip the hints since they were new to them in the given context. Baker [5] reviewed several reasons for why students game the system. While no clear answer was given, the question is applicable here as well. Pintrich [9] suggests that while appropriate motivation facilitates the use of existing metacognitive skills, other motivations may hinder such productive behavior.

Motivational issues bring us to our final hypothesis. Time Preference Discount is a term coined in economics that describes behavior in which people would rather have a smaller immediate reward over a distant greater reward. In the tutoring environment, comparing the benefit of immediate correct answer with the delayed benefit (if any) of acting in a metacognitively correct manner may often lead the student to choose the first. If that is indeed the case, then students may already have the right metacognitive

skills in place. The question we should be asking ourselves is not only how to get students to learn the desired metacognitive skills – but mainly, how to get students to use them.

Acknowledgements. We thank Ryan deBaker, Jo Bodnar, Ido Jamar, Terri Murphy, Sabine Lynn, Dale Walters, Kris Hobaugh and Christy McGuire for their help. This work was supported by the Pittsburgh Science of Learning Center which is funded by the National Science Foundation (#SBE-0354420), by the Graduate Training Grant awarded by the Department of Education (#R305B040063), and by NSF (#IIS-0308200).

References

[1] Aleven, V., McLaren, B.M., Roll, I., & Koedinger, K.R. (2006). Toward meta-cognitive tutoring: A model of help seeking with a Cognitive Tutor. *International Journal of Artificial Intelligence in Education* a(16), 101-30

[2] Aleven, V., Roll, I., McLaren, B.M., Ryu, E.J., & Koedinger, K.R. (2005) An architecture to combine meta-cognitive and cognitive tutoring: Pilot testing the Help Tutor. *in proceedings of 12Th International Conference on Artificial Intelligence in Education.* The Netherlands: IOS press.

[3] Aleven, V., Stahl, E., Schworm, S., Fischer, F., & Wallace, R.M. (2003). Help Seeking and Help Design in Interactive Learning Environments. *Review of Educational Research* 73(2), 277-320

[4] Baker, R.S., Corbett, A.T., & Koedinger, K.R. (2004) Detecting Student Misuse of Intelligent Tutoring Systems. *in proceedings of 7Th International Conference on Intelligent Tutoring Systems,* 531-40.

[5] Baker, R.S., Roll, I., Corbett, A.T., & Koedinger, K.R. (2005) Do Performance Goals Lead Students to Game the System? *in proceedings of 12Th International Conference on Artificial Intelligence in Education,* 57-64. Amsterdam, The Netherlands: IOS Press.

[6] Chang, K.K., Beck, J.E., Mostow, J., & Corbett, A. (2006) Does Help Help? A Bayes Net Approach to Modeling Tutor Interventions. *in proceedings of Workshop on Educational Data Mining at AAAI 2006,* 41-6. Menlo Park, California: AAAI.

[7] Gama, C. (2004) Metacognition in Interactive Learning Environments: The Reflection Assistant Model. *in proceedings of 7Th Conference on Intelligent Tutoring Systems,* 668-77. Berlin: Springer-Verlag.

[8] Koedinger, K.R., & Anderson, J.R. (1997). Intelligent Tutoring Goes to School in the Big City. *International Journal of Artificial Intelligence in Education* 8, 30-43

[9] Pintrich, P.R. (1999). The role of motivation in promoting and sustaining self-regulated learning. *International Journal of Educational Reserach* (31), 459-70

[10] Reif, F., & Scott, L.A. (1999). Teaching scientific thinking skills: Students and computers coaching each other. *American Journal of Physics* 67(9), 819-31

[11] Roll, I., Aleven, V., McLaren, B.M., Ryu, E., Baker, R.S., & Koedinger, K.R. (2006) The Help Tutor: Does Metacognitive Feedback Improves Students' Help-Seeking Actions, Skills and Learning? *in proceedings of 8Th International Conference in Intelligent Tutoring Systems,* 360-9. Berlin: Springer Verlag.

[12] Roll, I., Baker, R.S., Aleven, V., McLaren, B.M., & Koedinger, K.R. (2005) Modeling Students' Metacognitive Errors in Two Intelligent Tutoring Systems. *in proceedings of User Modeling 2005,* 379-88. Berlin: Springer-Verlag.

[13] Roll, I., Ryu, E., Sewall, J., Leber, B., McLaren, B.M., Aleven, V., & Koedinger, K.R. (2006) Towards Teaching Metacognition: Supporting Spontaneous Self-Assessment. *in proceedings of 8Th International Conference on Intelligent Tutoring Systems,* 738-40. Berlin: Springer Verlag.

[14] Schworm, S., & Renkl, A. (2002) Learning by solved example problems: Instructional explanations reduce self-explanation activity. *in proceedings of The 24Th Annual Conference of the Cognitive Science Society,* 816-21. Mahwah, NJ: Erlbaum.

[15] Tobias, S., & Everson, H.T. (2002). Knowing what you know and what you don't: further research on metacognitive knowledge monitoring. *College Board Research Report 2002-2003*

[16] White, B.Y., & Frederiksen, J.R. (1998). Inquiry, Modeling, and Metacognition: Making Science Accessible to All Students. *Cognition and Instruction* 16(1), 3-118

[17] Wood, H.A., & Wood, D.J. (1999). Help seeking, learning, and contingent tutoring. *Computers and Education* 33(2), 153-69

[18] Yudelson, M.V., Medvedeva, O., Legowski, E., Castine, M., Jukic, D., & Crowley, R.S. (2006) Mining Student Learning Data to Develop High Level Pedagogic Strategy in a Medical ITS. *in proceedings of Workshop on Educational Data Mining at AAAI 2006,* Menlo Park, CA: AAAI.

ITS, Feedback and Scaffolding

This page intentionally left blank

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Using Item Response Theory (IRT) to select hints in an ITS

MICHAEL J. TIMMS
WestEd, 300 Lakeside Drive, 25th Floor, Oakland, CA 94612-3534, USA
(510) 302 4214, mtimms@wested.org

Abstract. Many Intelligent Tutoring Systems (ITSs) are designed to tutor students by providing error feedback and hints as students solve a series of problems, where the tutoring system's feedback is based on a model of the student's ability in the subject domain. Few ITSs, however, model the difficulty of the problems the student is solving, and those that do rarely utilize an empirical approach. Item Response Theory, a methodology applied in educational measurement, offers a way to empirically model both a student's ability and the difficulty of the problem on a common scale. This paper first describes a method for using Item Response Theory (IRT) to model the difficulty of the problems (items), which allows the system to determine the level of hints that students need during problem solving. Second, the paper reports on how the method was used in the FOSS Self-assessment System, an ITS developed to accompany part of a middle school science curriculum module. Finally, the paper presents the results of a study that compared three versions of the Self-assessment System to evaluate its efficacy, which showed that the methodology holds some promise, but needs further refinement and testing.

Introduction

Many Intelligent Tutoring Systems (ITSs) are designed to tutor students as they work gradually through a problem, with the ultimate goal that students eventually gain the skills and knowledge necessary to solve the problems unaided. ITSs usually have well-developed methods for modeling a student's knowledge and skills, but surprisingly few systems have equally well-developed methods for modeling the difficulty of the problems being posed by the tutor. Ignoring item difficulty is a major omission because the difficulty of the problem has a large effect on how productive the interaction between the student and the learning materials will be. If the student finds the problem too easy, little learning will occur. In contrast, if the problem is too difficult for the student to complete successfully, they will learn nothing and may also become discouraged.

Systems that do take account of the difficulty of the problems posed to the student include Ecolab [5], AnimalWatch [1], and DynaMath [3] but these are in the minority. Ecolab, AnimalWatch and Dynamath each approached the issue by constructing a definition of Vygotsky's Zone of Proximal Development [7, 8] and optimizing student problem solving so that students were always interacting with problems at a level of difficulty that would be productive for them given appropriate help from the system. In Ecolab [5] students were offered a level of help that was reflective of the complexity and abstraction of the learning material they were interacting with. AnimalWatch [1] applied a definition of the Zone of Proximal Development (ZPD) based upon the number of hints that learner requests per problem over a series of problems. Murray and Aroyo called the Specific ZPD

(akin to specific heat in chemistry) or SZPD. The SZPD comprised three parameters. H, the target number of hints in each problem set; DH, the allowable variation in H that will still keep the student in the ZPD; and P, the minimum number of hints a student is guaranteed to see under normal circumstances. The values of H, DH and P are determined empirically by trying out the system until the desired teaching and learning style is accomplished. DynaMath [3] successfully used knowledge from a pre-test to set problems that are consistently in the student's Zone of Proximal Development. DynaMath pre-tested students on their multiplication tables with a set of 100 questions before they used the tutor to learn how to do multi-digit multiplication. The system used a simplistic estimate of the difficulty of each multi-digit multiplication problem by predicting it based upon the number of digits in the numbers being multiplied. For example, a multiplication problem in the form "n × nn" (e.g., 6 × 22) is easier than one of the form "nn × nn" (e.g., 66 × 22). Knowing this, and knowing from the pre-test data which parts of the multiplication tables the student had mastered, the tutor constructs a unique set of 80 problems for an individual student. The 80 problems are administered in a sequence that gradually increases the difficulty for that particular student. DynaMath gives progressive levels of assistance to students as they make mistakes, gradually moving them on through the 80 questions.

Each of these systems developed a unique method for judging item difficulty when, in fact, there are empirical methods that have been widely used in educational measurement for many years to accurately measure item difficulty. This paper describes one such method, Item Response Theory (IRT), and shows how it was used to model the difficulty of the problems (items), thus allowing the tutoring system to determine the level of hints that students need during problem solving. Next, the paper reports on how the method was used in the FOSS Self-assessment System, an ITS developed to accompany part of a middle school science curriculum module. Finally, the paper presents the results of a study that compared three versions of the Self-assessment System to evaluate its efficacy.

IRT is a well-established methodology in the field of educational measurement, where one of its best-known uses is to select test items in computer adaptive tests, assessments that essentially mold themselves to the ability of the examinee. VanLehn [6] notes that IRT has powerful features, including the ability to empirically determine the difficulty of items, but work is needed to take advantage of this in tutoring systems. IRT is also able to model the ability of the student. However, to date, very few ITSs have made use of IRT as a method of estimating item difficulty, let alone for estimating student ability [4]. This is unfortunate because IRT models offer the flexibility to model several features of item difficulty and, of particular interest for ITSs, to simultaneously model the learner's subject matter mastery. For example, the Rasch item response model, one of simplest IRT measurement models, is particularly well suited to analyses of item difficulty in ITSs because many of them pose problems in which the response, or steps in a response, can be classified as correct or incorrect. In the simple Rasch model, the probability that a student will give a correct response to a problem with a right/wrong answer format is conditional on the ability of the student and the difficulty of the item. The simple Rasch model follows a logistic response function and may be written as:

$$\Pr[X_{ij} = 1 \mid \theta_i, \beta_j] = \frac{\exp(\theta_i - \beta_j)}{1 + \exp(\theta_i - \beta_j)} \quad \text{and} \quad \Pr[X_{ij} = 0 \mid \theta_i, \beta_j] = \frac{1}{1 + \exp(\theta_i - \beta_j)}. \tag{1}$$

Where, $X_{ij} = 1$ or 0 indicates that student i answered problem j correctly or incorrectly, respectively, θ_i = ability of student i and, β_j = difficulty of problem j.

The probability that a student will answer a question correctly can be specified as a function of the difference between the ability of the student and the difficulty of a question. Both ability and difficulty are measured on the same linear scale using logits, a unit based on the log-of-the-odds. This means that the difference between a student's ability estimate and the item difficulty has a direct meaning for student performance [2], and they can be meaningfully compared. Say, for example, that a student has an estimated ability level of 1 logit. If that student faced a problem with a difficulty equal to 1 logit, then there would be a difference of 0 logits between the student ability and the item difficulty, which means that, according to equation [1] above, there is a 50% chance that the student will get the problem correct without help. But, if the same student encounters a problem that has a difficulty of 2 logits, then the student ability is one logit lower than the difficulty, so the probability that he will get it correct is reduced to about 27%. Similarly, if that student tackles a problem with a difficulty of -1 logits, which is 2 logits below his ability level, then there is an 88% likelihood that the student will be able to complete the problem successfully. So, knowing the pre-test estimates of ability of the students and the estimates of difficulty of the problems, the system can predict on which problems a student is most likely to need help.

1. The FOSS Self-assessment System ITS

To investigate how this feature of IRT could be used to deliver hints in an ITS, a self-assessment system was created in which students received feedback while they practiced solving science problems, and were also able to monitor their own progress through short summative assessments. The development work was part of the NSF-funded Principled Assessment Designs for Inquiry (PADI) project, which involved the University of Maryland, SRI International, and the Graduate School of Education and the Lawrence Hall of Science at the University of California, Berkeley (NSF grant REC-0129331).

The Self-assessment System was developed as a supplementary activity for a section of the Full Option Science System (FOSS) middle school Force and Motion curriculum that dealt with student understanding of speed and the application of the speed equation (velocity = distance/ time). The development team selected this relatively small portion of the curriculum because it had several physics problems for students to solve, but was not so large that it would be outside of the scope of the resources of the project to develop a computer based self-assessment system for it. The self-assessment system was designed to give students an opportunity to practice what they had learned about solving problems concerning speed, distance and time. The system was Internet-based, and contained a tutorial on how to use the system, a 10-item pretest, ten levels of problem types with a 5-item self-assessment between each level, and a final posttest (identical to the pretest). Figure 1 shows a flowchart of how students progressed through the system. In the pretest, the student answered the ten problems unaided by the system. The student's scores on each step of every problem were recorded and sent to the scoring engine, developed by the Berkeley Evaluation and Assessment Research (BEAR) center at UC Berkeley. The scoring engine used IRT to estimate the student ability based on the pretest. The ability estimate was sent to the hints system, which calculated the ability gap between the student ability and the upcoming set of items and allocated a level of hints as shown in Table 1.

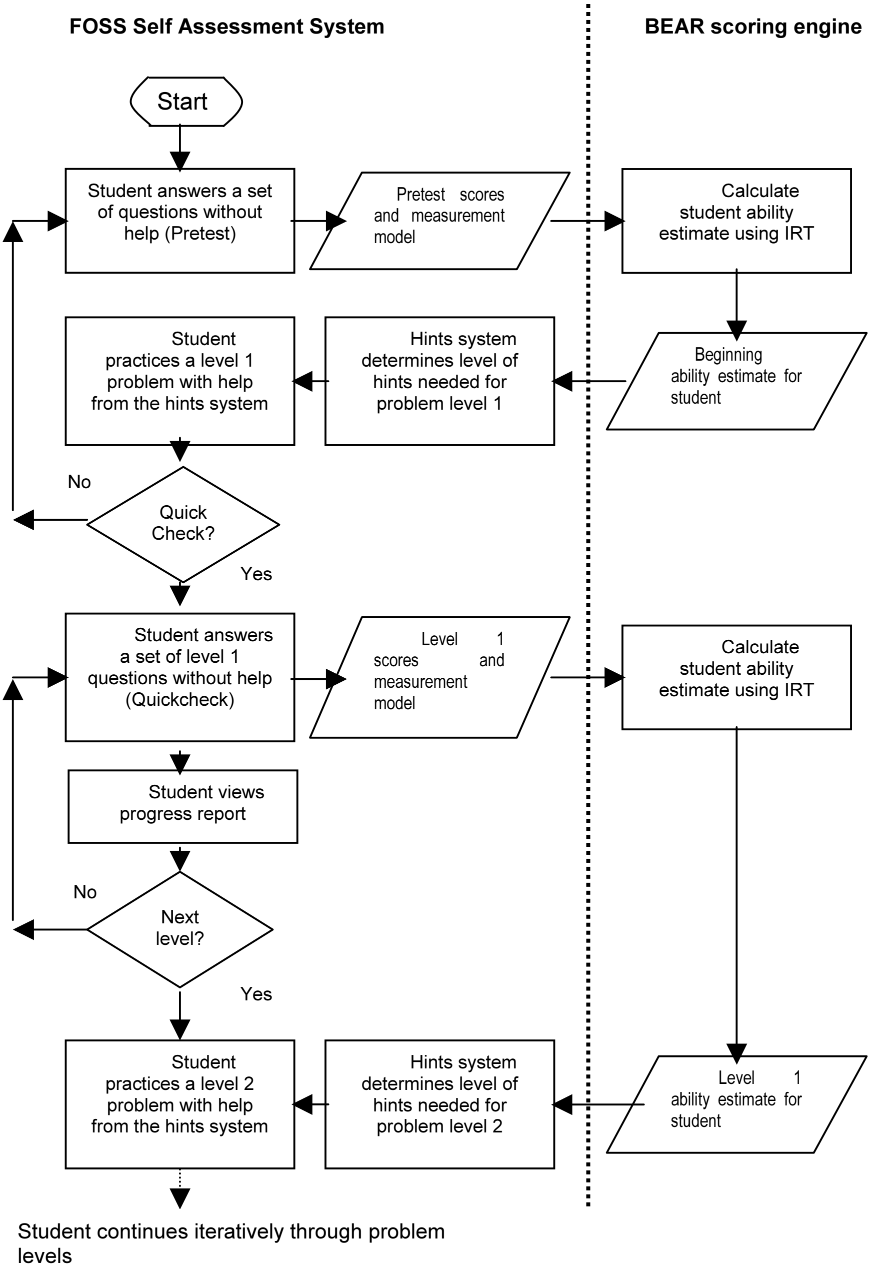


Figure 1. Flowchart of the Self-assessment System

Table 1. Level of hints given depending on ability gap.

Ability Gap (in logits)	Level of hints given	Description of hints level
-0.51 and below	3	Greatest level of help that points out actions necessary, labels equations, and gives values or calculation results. (With a -0.51 logit ability gap, the student has only a 37.5% chance of getting the problem correct unaided.)
-0.50 to – 0.01	2	Medium level of help that points out actions necessary but does not give values or calculation results. (With a -.01 logit ability gap, the student has a 50.25% chance of getting the problem correct unaided.)
0.0 and above	1	Lowest amount of help. Very small, prompting hints. (With a logit ability gap of 0.0 or above, the student has a 50% or greater chance of getting the problem correct unaided.)

The difficulty of the items at each level was estimated based on a paper-and-pencil pilot test of the items taken by 28 subjects. Based on the ability gap, the hints engine classified the student as needing one of three levels of hints for the next level of problems. Table 2 shows an example of the three levels of hints.

Following the pretest, students could practice on a level 1 problem, which involved calculating the speed of an arrow given distance and time. Figure 2 shows a screenshot of a typical problem from the practice session. Students selected the equation necessary to solve the problem. The system would intervene and tell them if they selected the wrong equation. Then, students completed the problem by entering the relevant values of distance and time from the information in the problem statement and diagram, and calculated the result. At any point, they could press the "check my work" button, which caused the system to check the sequence of steps in the problem for correctness. For each incorrect step, the system gave them error feedback and hints, and continued to do so for each incorrect step until the student made corrections so that there were no more errors. Based on their calculated ability gap (as measured by IRT), students were categorized as needing one of the three levels of hints as in Table 2, which shows an example of the three types of hints for just one step in a problem. There were specific hints for each problem step.

Table 2. Example of three levels of hints for a step in the problem

Hint level 3 (For large ability gap)	Hint level 2 (For medium ability gap)	Hint level 1 (For small ability gap)
You have entered the right numbers, but the units are wrong. The distance is in meters and should be written as 200m. Enter that. Time is in seconds and should be written as 25s. Enter that now.	You have entered the right numbers, but the units are wrong. The distance is in meters. Time is in seconds. Enter the units now.	You have entered the right numbers, but the units are wrong. Look in the problem. Enter the units now.

Students then received the hints they needed from their particular assigned hint level, which remained constant for that problem. As shown in Table 1, for students whose ability gap was large, a level 3 hint was delivered. The level 3 hints offered the greatest amount of help

and were given when the system calculated that the student had only a 37.5% chance of getting the problem correct unaided. A level 3 hint pointed out actions necessary, labeled equations, and gave values or calculation results. Level 2 hints were given when the system calculated using IRT that the student had only a 50.25% chance of getting the problem correct unaided. Level 2 hints pointed out actions necessary but did not give values or calculation results. The least detailed hints, in Level 1, were given when the system determined that the student had a 50% or greater chance of getting the problem correct unaided. Level 1 hints only prompted the student to think again about what the error might be.

Practice activity

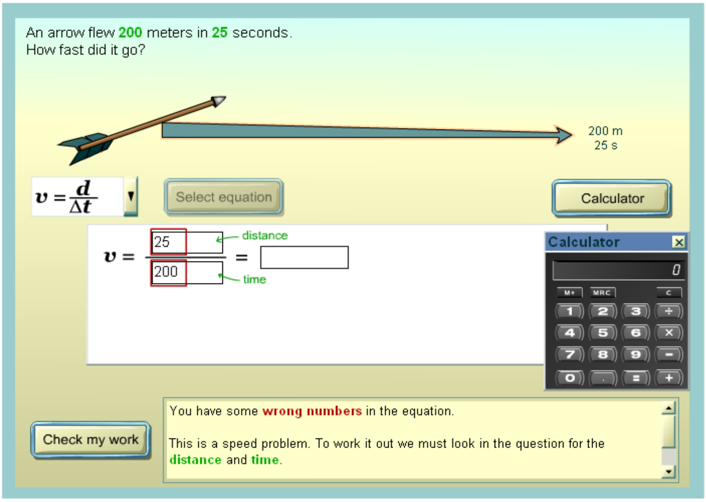


Figure 2. Screen shot of a second hint for problem level 1

Once a student had successfully completed at least one problem in the practice session, they could choose to take a “quick check” assessment of between 3-5 items. After the quick check, they could see a report of how they had performed. They then chose to either practice more, or go on to the next level. Students proceeded like this through all ten levels, a process that took several days and was paced with the curriculum, which introduced more and more complex problems as it went on. At the end of the last practice session at level 10, the students took the posttest, which comprised the same ten items they had taken in the pretest.

2. Empirical study

A study was implemented to determine first if IRT is an effective method to use in an ITS to determine what level of hint should be given to a student, and second, what effect an IRT based help system might have on student learning. For the study, three versions of the Self-assessment System were created. Version 1 (Full Tutor) operated as described so far, giving error feedback and hints in the practice sessions based on the ability gap as measured by IRT. Version 2 (Error Feedback) only highlighted of errors where there was a wrong or missing answer and gave following standard message in the hint text box: “You have made

a mistake in the area that is marked in red. Please correct it and submit your answer again." In version 3 (No Help) there was no feedback or hints in the practice activity section of the self-assessment system. Students simply were able to practice the types of questions before opting to take the quick check item set. Twenty classes of students were randomly assigned to one of the three experimental groups, each of which used a different version of the self-assessment system. Table 3 summarizes the allocation of students to the three experimental conditions.

Table 3. Experimental conditions

	Experimental Group		
	1 (n=160 of whom 101 completed sufficient levels to measure)	2 (n=125 of whom 81 completed sufficient levels to measure)	3 (n=52 of whom 31 completed sufficient levels to measure)
10-item pretest/posttest	Yes	Yes	Yes
Help during practice items	Yes	Yes	No
Type of help in practice	Error feedback and hints based on ability gap	Error feedback only	None
Quick-check assessments between levels	Yes	Yes	Yes

3. Results

As noted in Table 3, not many students completed the pretest, all ten levels of practice problems and quick checks, and the posttest. This was particularly so in experimental group 3 (No Help) in which very few students completed the whole of the study. This is not surprising, since that group received no help from the system during the practice problem solving sessions. The original intent had been to measure learning gains between pretest and posttest, but not many completed both. So, instead, learning gains were measured as the difference in logits between students' pretest ability estimates and their average ability across all the quick check assessments they completed and the posttest, if completed. This seemed a valid alternative because all the measures of ability were from assessments in which the student received no tutorial help, no matter what experimental group they were in. Also, each quick check ability estimate was based on performance of between 3 to 5 questions, so using several quick checks as a measure of ability meant that the estimate of ability was based on many items.

A comparison of the mean learning gains of students in the three experimental groups revealed statistically significant differences between the mean learning gains of both the Full Tutor and the Feedback Only groups when compared to the No Help group. As shown in the table and graph in Figure 3, the gains for the Full Tutor and Error Feedback groups were .98 and .77 logits higher, respectively, than the No Help group, which represented moderate standardized effect sizes of .70 and .67. These differences were educationally significant too, since the size of the difference in logits was almost the same as the spread of item difficulty between the easiest and the most difficult problems on the 10-item posttest.

Experimental Group	Mean (Logits)	N	Standard Deviation
1. Full Tutor	2.80	101	1.40608
2. Feedback Only	2.58	80	1.00781
3. No Help	1.81	31	1.40639
Total	2.57	212	1.30668

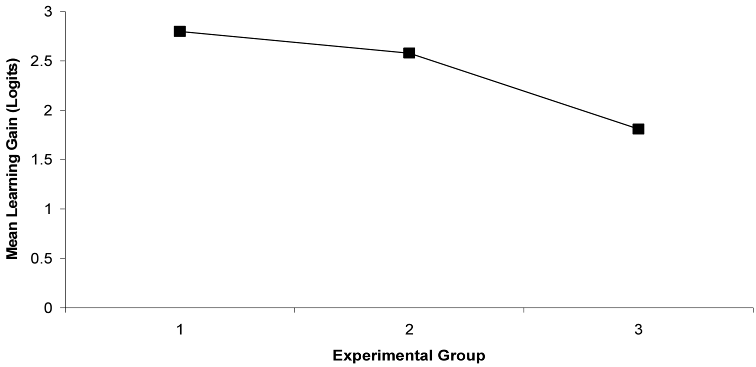


Figure 3. Mean Learning Gains for Each Experimental Group

4. Discussion

These findings demonstrated several things to me. First, the study showed that it is possible to build an intelligent help system that uses IRT to deliver hints to students according to their needs. The system technically worked well, demonstrating the possibility of using web-based ITSs with a centralized scoring engine that is accessed via the Internet. Second, it showed that such an IRT-based intelligent hints system can produce learning gains in the medium effect size range when compared to a system that provided neither error feedback nor hints. The result that the Full Tutor with the IRT based help system produced learning gains that were significantly larger than those produced by the No Help version was expected. However, the fact that the Full Tutor did not also perform significantly better than the Error Feedback version was unexpected. Further investigation revealed two factors that most likely account for this. First, the subject domain turned out to be easier for students to master than had been anticipated. Students in the Full Tutor group had their ability estimate updated after each of the ten problem levels. Examination of the data revealed that at the first problem level, the bulk of students (62%) were categorized as needing the most helpful hints. By the second problem level, 86% of students were classed as needing only the least level of help, and this pattern persisted through the remaining problem levels. As can be seen from Table 1, the type of hint received in that low-help category is mainly error identification and encouragement to reconsider the problem step. That feedback is almost identical to the help that the students in the Error Feedback experimental group received throughout the problem solving. So, for most of the time, students in the Full Tutor and Error Feedback tutor group received almost identical help, so it is hardly surprising that they performed about the same.

Another factor that may explain this result is that there were some possible inaccuracies in the item estimates. Item difficulties were set based on a paper and pencil pretest. In the

quick check items administered between problem-solving practice levels, the system generated items on-the-fly by varying values of distance, time and speed in the problems so that students had a variety of items each time. As the values used in problems were kept within certain parameters, an assumption was made that item difficulties would remain fairly stable. This assumption was incorrect in some places. Both of these factors may have affected the accuracy of the Full Tutor's decisions about which level of hints a student should receive.

Another weakness of the system was that, although it re-measured student ability at each problem level, it assumed fixed difficulty estimate values for items throughout, and did not take account of the fact that as ability increased over time. So, for example, a student who was taking a QuickCheck set of items that included items from the previous two levels would find them easier than she had when she first did them two levels earlier.

5. Conclusions and recommendations

The study showed that IRT could be a useful methodology to use in determining how much help students need in solving problems in an intelligent tutoring system, and demonstrated how a system might work. The methodology was applied in this experiment to items that had a single path correct solutions, but it could also be applied to other more complex problems where there are multiple paths to a correct solution However, there are some future design improvements that would likely improve the accuracy of the ITS. A future system should take account of the individual item difficulty of each variant of a problem, which will mean more pilot testing of the system to obtain reliable item statistics. These findings point to the need for a further study to correct some of the design weaknesses of the Full Tutor system in this study. In addition, a future study might tackle a more challenging domain of instruction so that there is enough depth to thoroughly test the differences between types of tutors. Finally, since some lower ability learners struggle with reading, it would be interesting to investigate if a system in which the hints are given verbally, as well as in writing, would be more beneficial to those learners.

References

[1] Arroyo, I., Beck, J. E., Woolf, B., Beal, C. R., & Schultz, K. (2000). *Macroadapting Animalwatch to gender and cognitive differences with respect to hint interactivity and symbolism.* Paper presented at the Fifth International Conference on Intelligent Tutoring Systems, Montreal, Canada.

[2] Embretson, S. E., & Reise, S. P. (2000). Item Response Theory for Psychologists. Mahwah, New Jersey: Lawrence Erlbaum Associates.

[3] Gerber, M. M., Semmel, D. S., & Semmel, M. I. (1994). Computer-Based dynamic Assessment of Multidigit Multiplication. *Exceptional Children, 61*(2), 114-125.

[4] Junker, B. (1999). *Some Statistical Models and Computational Methods that may be Useful for Cognitively-relevant Assessment*. Pittsburgh, PA: Carnegie Mellon University.

[5] Luckin, R., & du Boulay, B. (1999). Ecolab: The Development and Evaluation of a Vygotskian Design Framework. *International Journal of Artificial Intelligence in Education, 10*, 198-220.

[6] VanLehn, K. (2006) The behavior of tutoring systems. *International Journal of Artificial Intelligence in Education*. 16, 227-265.

[7] Vygotsky, L. S. (1978). *Mind in Society: the Development of Higher Psychological Processes.* Cambridge, MA: Harvard University Press.

[8] Vygotsky, L. S. (1986). *Thought and Language*. Cambridge, MA: The MIT Press.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

What Level of Tutor Interaction is Best?

Leena RAZZAQ, Neil T. HEFFERNAN and Robert W. LINDEMAN
Computer Science Department, Worcester Polytechnic Institute
100 Institute Road, Worcester, Massachusetts, USA

Abstract. Razzaq and Heffernan (2006) showed that scaffolding compared to hints on demand in an intelligent tutoring system could lead to higher averages on a middle school mathematics post-test. There were significant differences in performance by condition on individual items. For an item that proved to be difficult for all of the students on the pretest, an ANOVA showed that scaffolding helped significantly ($p < 0.01$). We speculated that the scaffolding had a greater positive effect on learning for this item because it was much more difficult for the students than the other items. We thought that this result warranted a closer look at the link between the difficulty of an item and the effectiveness of scaffolding. In this paper, we report on an experiment that examines the effect of math proficiency and the level of interaction on learning. We found an interesting interaction between the level of interaction and math proficiency where less-proficient students benefited from more tutor interaction and more-proficient students benefited from less interaction.

Keywords. Intelligent tutoring, scaffolding, feedback, interactive help, math tutoring

1. Introduction

Several studies in the literature have argued that human tutors that are more interactive lead to better learning and can achieve greater learning gains. In a comparison of Socratic and didactic tutoring strategies, Core, Moore and Zinn [1] found that the more interactive (based on words produced by students) Socratic tutorial dialogs had a greater correlation with learning. Katz, Connelly and Allbritton [2] found that students learned more when they participated in post practice dialogs with a tutor than students who did not. Chi, Siler, Jeong, Yamauchi and Hausmann [3] found that students who engaged in a more interactive style of human tutoring were "able to transfer their knowledge better than the students in the didactic style of tutoring."

We are interested in the role of tutor interaction in intelligent tutoring systems (ITS) and similar results have been found in other studies of interactive ITS. Razzaq and Heffernan [4] found a positive effect on learning when students worked on solving equations in an ITS called E-tutor which incorporated tutorial dialogs. E-tutor is a model-tracing tutor that is able to carry on a coherent dialog that consists of breaking down problems into smaller steps and asking new questions about those steps, rather than simply giving hints. Tutorial dialogs were chosen from transcripts of human tutoring sessions and were incorporated in E-tutor. E-tutor does not have a hint button and when students make errors they are presented with a tutorial dialog if one is available. The student must respond to the dialog to exit it and return to solving the original problem. Students stay in the dialog loop until they respond correctly or the tutor has run out of dialog. When the

tutor has run out of dialog, the last tutorial response presents the student with the correct action and input similar to the bottom-out hint in a hint sequence. A close mapping between the human tutor dialog and the E-tutor dialog was attempted. E-tutor was compared to a control version that did not engage in dialog, but did have a hint button to supply hints to students when they asked for them. E-tutor with dialog led to better learning and represents a more interactive tutor than the "*hints on demand*" control condition.

It seems that a positive relationship between learning and tutor interaction exists, and we would expect students to learn more whenever they engage in interactive tutoring conditions than in less interactive conditions such as reading text. There is, however, evidence that this is not always the case. VanLehn, Graesser, Jackson, Jordan, Olney, & Rose [5] reviewed several studies that hypothesize that the relationship between interactivity and learning exists, as well as a few studies that failed to find evidence for this relationship. VanLehn et al [5] found that when students found the material to be difficult, tutoring was more effective than having the students read an explanation of how to solve a problem. However, this was not the case when the students found the material to be at their level: interactive tutoring was not more effective than canned-text. We found a similar effect in a study in Razzaq and Heffernan [6].

We used the ASSISTment system [7], a web-based tutoring system that blends assisting students with assessing their knowledge, in this study. The system tutors students on 7th, 8th and 10th grade mathematics content that is based on Massachusetts Comprehensive Assessment System (MCAS) released test items. There are currently over 1000 students using the ASSISTment system as part of their mathematics class.

Our results show that students are learning 8th grade math at the computer by using the system [7], but we were not certain if this is due to students getting more practice on math problems or more due to the intelligent tutoring that we created. Students are forced to participate in this intelligent tutoring if they get a problem incorrect. In Razzaq and Heffernan [6], we conducted a simple experiment to see if students learned on a set of 4 problems if they were forced to do the scaffolding questions, which would ASK them to complete each step required to solve a problem, compared with being given hints on demand, which would TELL them the same information without expecting an answer to each step. In the study, the "*scaffolding + hints*" condition represents a more interactive learning experience than the "*hints on demand*" condition.

The results of the Razzaq and Heffernan [6] experiment showed that *scaffolding + hints* led to higher averages on a post-test than *hints on demand*, although it was not statistically significant. When we compared scores on particular post-test items that students had seen as pretest items, we found significant differences by condition. For one item, which concerned finding the y-intercept from an equation, the ANOVA showed a statistically significant ($p < 0.01$) with an effect size of 0.85. This item on finding the y-intercept from an equation proved to be a difficult problem for all of the students on the pretest and scaffolding helped significantly. We speculated that the *scaffolding + hints* had a greater positive effect on learning for the first pretest item because it was much more difficult for the students than the second pretest item. We thought that this result warranted a closer look at the link between the difficulty of an item and the effectiveness of scaffolding.

In this study, we look at three different conditions, in the ASSISTment system, which have varying levels of tutor interaction. The first two conditions are the same as in [6] (*scaffolding + hints* and *hints on demand*). The third condition is a *delayed feedback* condition where students get no feedback from the tutor until they finish all of the

problems in the experiment, whereupon they receive worked out solutions to all of the problems. Both immediate and delayed feedbacks have been shown to be helpful to students (Mathan and Koedinger) [8]. In Razzaq and Heffernan [4], our human tutor provided immediate feedback to student errors, on most occasions, keeping students on the correct solution path. There was one out of 26 problems where the human tutor gave delayed feedback to a student error. In this instance, the tutor allowed the student to continue where she had made an error and then let her check her work and find the error seemingly promoting evaluative skills. This happened with the student who was taking a more advanced algebra class and was slightly more advanced than the other students in the study. However, for the other 25 problems, the tutor provided immediate feedback to promote the development of generative skills. This agrees with McArthur et al [9] in their examination of tutoring techniques in algebra where they collected one and a half hours of videotaped one-on-one tutoring sessions. "In fact, for every student error we recorded, there was a remedial response. At least the tutors we observed were apparently not willing to let students explore on their own and perhaps discover their own errors…teachers may believe that such explorations too frequently lead to unprofitable confusion for the student."

The purpose of this experiment was to determine which level of interaction worked best for students learning math: *scaffolding + hints*, *hints on demand* or *delayed feedback*, and how their math proficiency influenced the effectiveness of the feedback provided.

2. The ASSISTment System

Limited classroom time available in middle school mathematics classes requires teachers to choose between time spent assisting students' development and time spent assessing their abilities. To help resolve this dilemma, assistance and assessment are integrated in a web-based system called the ASSISTment [1] System which offers instruction to students while providing a more detailed evaluation of their abilities to teachers than is available under most current approaches. Many teachers use the system by requiring their students to work on the ASSISTment website for about 20 minutes per week in their schools' computer labs. Each week when students work on the website, the system "learns" more about the students' abilities and thus, it can hypothetically provide increasingly accurate predictions of how they will do on a standardized mathematics test [10]. The ASSISTment System is being built to identify the difficulties individual students - and the class as a whole – are having. It is intended that teachers will be able to use this detailed feedback to tailor their instruction to focus on the particular difficulties identified by the system. Unlike other assessment systems, the ASSISTment technology also provides students with intelligent tutoring assistance while the assessment information is being collected. The hypothesis is that ASSISTments can do a better job of assessing student knowledge limitations than practice tests by taking the amount and nature of the assistance that students receive into account.

It is easy to carry out randomized controlled experiments in the ASSISTment System [11]. Items are arranged in modules in the system. The module can be conceptually subdivided into two main pieces: the module itself, and sections. The

[1] The term ASSISTment was coined by Kenneth Koedinger and blends Assisting and Assessment.

module is composed of one or more sections, with each section containing items or other sections. This recursive structure allows for a rich hierarchy of different types of sections and problems. The section component is an abstraction for a particular listing of problems. This abstraction has been extended to implement our current section types, and allows for future expansion of the module unit. Currently existing section types include "Linear" (problems or sub-sections are presented in linear order), "Random" (problems or sub-sections are presented in a pseudo-random order), and "Choose One" (a single problem or sub-section is selected pseudo-randomly from a list, the others are ignored).

3. Experimental Design

Problems in this experiment addressed the topic of interpreting linear equations. Figure1 shows an item used in the experiment. The item shows the different feedback that students can receive once they have answered a question incorrectly. (We call this top-level question the original question.)

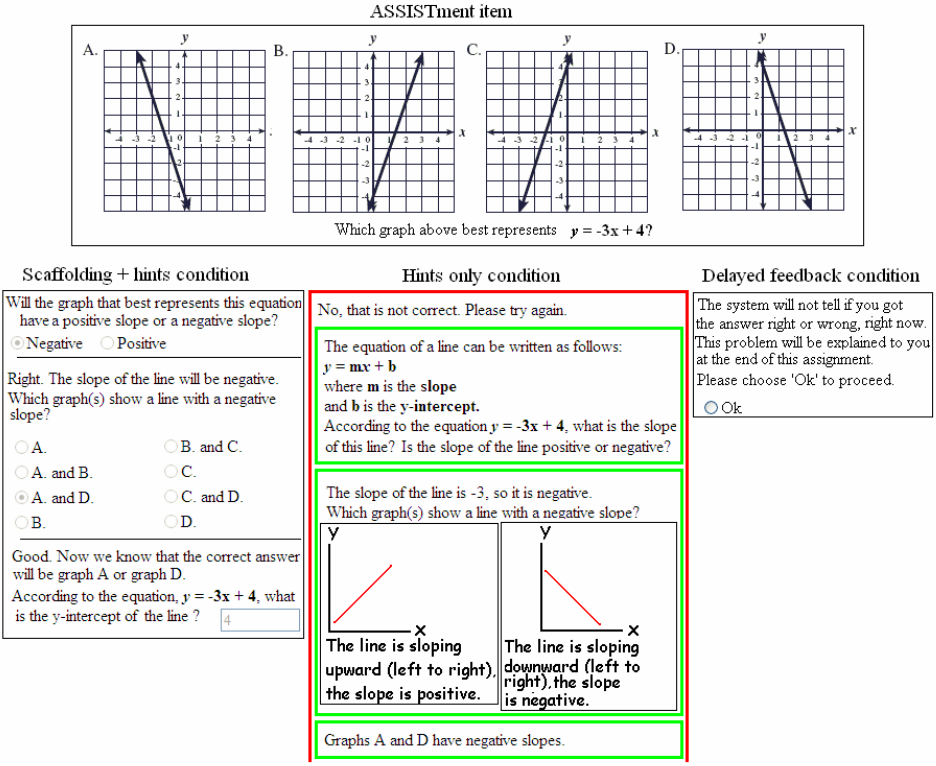


Figure 1. An ASSISTment item showing 3 different levels of interaction.

A student in the *scaffolding + hints* condition is immediately presented with the first scaffolding question. Students must answer a scaffolding question correctly to proceed and receive the next scaffolding question (or finish the problem). Students can

ask for hints on the scaffolding questions, but not on the original question. They cannot go back and answer the original question, but rather are forced to work through the problem.

Students in the hints condition receive a message, outlined in red, of "No, that is not correct. Please try again." The hints, outlined in green, appear when the student requests them by pressing the Hint button. Students do not see the hints unless they ask for them. Figure 1 shows a sequence of three hints to solving the problem. The full sequence has seven hints in total, with a bottom-out hint at the end of the sequence. The bottom-out hint gives the student the answer to the problem.

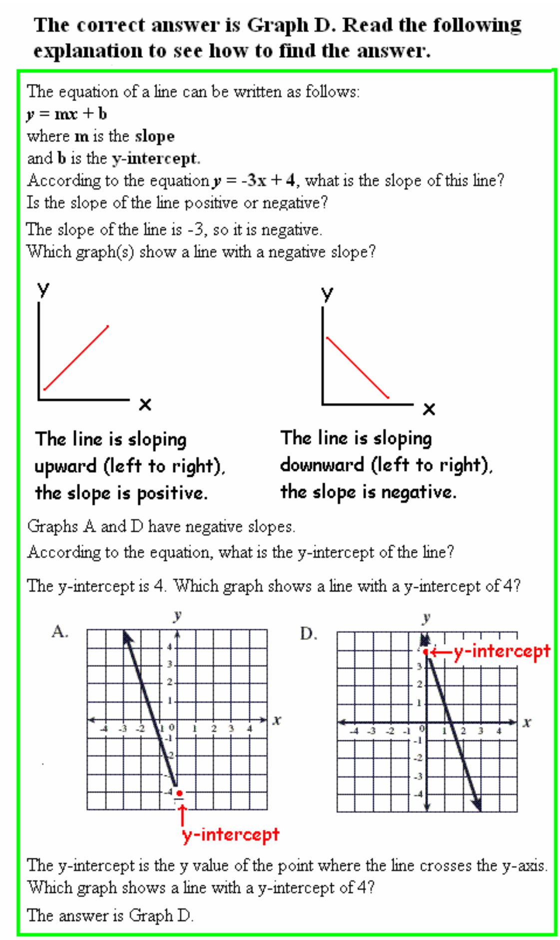


Figure 2. The delayed feedback condition provided explanations at the end of the assignment.

Students in the *delayed feedback* condition did not receive any feedback on the problems that they did until they had finished all of the problems. At that time, the students were presented with the answers and explanations of how to solve the problems. Figure 2 shows the explanation that students in the *delayed feedback* condition received for the item shown in Figure 1.

Based on the results of Razzaq and Heffernan [6], we hypothesized that less-proficient students would need more interaction and benefit more from the scaffolding than more-proficient students.

For this experiment, the number of problems was held constant, but students took as much time as they needed to finish all of the problems. Students were presented with two pretest problems, four experiment problems and four post-test problems that addressed the topic of interpreting linear equations. There were three versions of the experiment problems, one for each condition. Two of the pretest problems were repeated in the post-test.

The ASSISTment system randomly assigned students to the *scaffolding + hints*, *hints on demand* or *delayed feedback* conditions with equal probability. There were 366 eighth grade students from the Worcester Public Schools in Worcester, Massachusetts who participated in the experiment: 131 students were in honors level classes and 235 were in regular math classes. There were 119 students in the *scaffolding + hints*

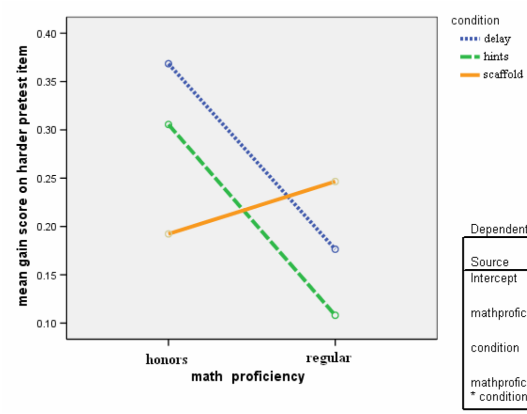
condition, 124 students in the *hints on demand* condition and 123 students in the *delayed feedback* condition. The students worked on the problems during their regular math classes.

4. Analysis

We excluded students who got all of the pretest problems correct from the analysis because we assumed that they knew the material. Fifty-one students got perfect scores on the pretest and were excluded. We first checked to make sure that the groups were not significantly different at pretest by doing an ANOVA on pretest averages by condition. There was no significant difference between groups at pretest (p = 0.556). Students learned overall from pretest to post-test (p = 0.005).

When we look at performance on the post-test by condition, the difference is not significant; however there is an interesting trend when we separate students by math proficiency. There is a significant interaction (p = 0.045) between condition and math proficiency on the post-test average. The regular students seem to benefit more from the *scaffolding* + *hints* condition, while honors students seem to benefit more from the *delayed feedback* condition. We also ran an ANOVA on the gain score of the two pretest items. Again, there is no statistically significant difference by condition, but there is a significant interaction between math proficiency and condition (p = 0.036), where once more, honors students do best in the *delayed feedback* condition and regular students do best in the *scaffolding* + *hints* condition.

We decided to take a closer look at the item that proved most difficult for students. The problem concerned finding the y-intercept from an equation and was presented to students in the pretest and again in the post-test. We did a one-way ANOVA using math proficiency as a covariate. An interaction between condition and math proficiency is evident (p = 0.078); honors students performed best when they received *delayed feedback* and the regular students performed best when they received *scaffolding* + *hints*.



Descriptive Statistics

Dependent Variable: gainscore_y_intercept

mathproficiency	condition	Mean	Std. Deviation	N
honors	d	.3684	.48885	38
	h	.3056	.46718	36
	s	.1923	.40192	26
	Total	.3000	.46057	100
regular	d	.1765	.38405	68
	h	.1081	.31264	74
	s	.2466	.43400	73
	Total	.1767	.38234	215

Tests of Between-Subjects Effects

Dependent Variable: gainscore_y_intercept

Source		Type III Sum of Squares	df	F	Sig.
Intercept	Hypothesis	14.522	1	124.897	.007
	Error	.237	2.039		
mathproficiency	Hypothesis	.835	1	1.968	.295
	Error	.853	2.010		
condition	Hypothesis	.232	2	.272	.786
	Error	.852	2		
mathproficiency * condition	Hypothesis	.852	2	2.577	.078
	Error	51.099	309		

Figure 3. There are significant interactions between condition and math proficiency.

5. Discussion

We interpret the low p-values on the interaction term to mean that there are different rates of learning on the single items based upon the interaction between the level of math proficiency and condition. Students who come in with less knowledge benefit more from the *scaffolding + hints* than students who come in with more knowledge. Students who come in with more knowledge benefit from the *delayed feedback* more than the other groups.

The results of this experiment were surprising. We did find evidence to support the interaction hypothesis for regular students. The regular students performed best in the *scaffolding + hints* condition, which is the most interactive condition. We did not expect students in the *delayed feedback* condition to learn more than in other groups, however, the honors students did better in this condition than in the *scaffolding + hints* or *hints on demand* conditions.

One possible explanation is that less-proficient students benefit from more interaction and coaching through each step to solve a problem while more-proficient students benefit from seeing problems worked out and seeing the big picture. Another possible explanation, put forth by one of the eighth grade teachers, is that honors students are often more competitive and like to know how they do on their work. The *delayed feedback* group had to wait until the end of the assignment to see whether they got the questions right or wrong. Perhaps the honors students ended up reading through the explanations more carefully than they would have read the scaffolding questions or hints because they were forced to wait for their results.

Chi [12] found a difference in the way that students used worked examples based on their proficiency in problem-solving. "… we find that the Good students use the examples in a very different way from the Poor students. In general, Good students, during problem solving, use the examples for a specific reference, whereas Poor students reread them as if to search for a solution." Although we did not present worked examples to the students in the *delayed feedback* condition during the experiment, the "worked solutions" to the experiment problems may have behaved as worked examples to the post-test problems.

The students in the *hints on demand* condition did not perform as well as the *delayed feedback* groups for both more-proficient and less-proficient students. One possible explanation is the more proactive nature of the *delayed feedback* explanations. Murray and VanLehn [13] found that proactive help was more effective for some students. "Proactive help when a student would otherwise flounder can save time, prevent confusion, provide valuable information at a time when the student is prepared and motivated to learn it, and avoid the negative affective consequences of frustration and failure." In the ASSISTment system, students only see hints if they ask for them and they are less likely to ask for hints on multiple choice questions when they can guess more easily.

We believe the results of this experiment provide further evidence of the interaction hypothesis for less-proficient students. However, the interaction between condition and math proficiency presents a good case for tailoring tutor interaction to types of students to maximize their learning.

Acknowledgements

We would like to thank all of the people associated with creating the ASSISTment system listed at www.ASSISTment.org including investigators Kenneth Koedinger and Brian Junker at Carnegie Mellon. We would also like to acknowledge funding from the US Department of Education, the National Science Foundation, the Office of Naval Research and the Spencer Foundation. All of the opinions expressed in this paper are those solely of the authors and not those of our funding organizations.

References

[1] Core, M. G., Moore, J. D., and Zinn, C. The Role of Initiative in Tutorial Dialogue, the 10th Conference of the European Chapter of the Association for Computational Linguistics, Budapest, Hungary, April, 2003.

[2] Katz, S., Allbritton, D., Connelly, J. Going beyond the problem given: How human tutors use post-solution discussions to support transfer. International Journal of Artificial Intelligence in Education 13 (2003) 79-116.

[3] Chi, M. T. H., Siler, S., Jeong, H., Yamauchi, T., & Hausmann, R. G. (2001). Learning from tutoring. *Cognitive Science*, 25:471-533.

[4] Razzaq, L. & Heffernan, N. T. (2004) Tutorial dialog in an equation solving intelligent tutoring system. In J. C. Lester, R. M. Vicari, & F. Parguacu (Eds.) Proceedings of 7th Annual Intelligent Tutoring Systems Conference, Maceio, Brazil. Pages 851-853.

[5] VanLehn, K., Graesser, A. C., Jackson, G. T., Jordan, P., Olney, A., & Rose, C. P. When is reading just as effective as one-on-one interactive human tutoring? In Proceedings of the 27th Annual Meeting of the Cognitive Science Society (pp. 2259-2264). Mahwah, NJ: Erlbaum. (2005) 2259-2264.

[6] Razzaq, L., Heffernan, N. T. (2006). Scaffolding vs. hints in the ASSISTment System. In Ikeda, Ashley & Chan (Eds.). *Proceedings of the 8th International Conference on Intelligent Tutoring Systems*. Springer-Verlag: Berlin. pp. 635-644. 2006.

[7] Razzaq, L., Heffernan, N. T., Koedinger, K. R., Feng, M., Nuzzo-Jones, G., Junker, B. Macasek, M. A., Rasmussen, K. P., Turner, T. E., & Walonoski, J. A. (2007). Blending Assessment and Instructional Assistance. In Nadia Nedjah, Luiza deMacedo Mourelle, Mario Neto Borges and Nival Nunesde Almeida (Eds). Intelligent Educational Machines within the Intelligent Systems Engineering Book Series. 23-49 Springer Berlin / Heidelberg.

[8] Mathan, S. & Koedinger, K. R. (2003). Recasting the Feedback Debate: Benefits of Tutoring Error Detection and Correction Skills. In Hoppe, Verdejo & Kay (Eds.), *Artificial Intelligence in Education: Shaping the Future of Learning through Intelligent Technologies. Proceedings of AI-ED 2003* (pp. 39-46). Amsterdam, IOS Press.

[9] McArthur, D., Stasz, C., & Zmuidzinas, M. (1990) Tutoring techniques in algebra. *Cognition and Instruction*. 7 (pp. 197-244.)

[10] Feng, M., Heffernan, N. T., Koedinger, K. Predicting State Test Scores Better with Intelligent Tutoring Systems: Developing Metrics to Measure Assistance Required. Submitted to the 8th International Conference on Intelligent Tutoring Systems (2005).

[11] Nuzzo-Jones, G., Walonoski, J. A., Heffernan, N. T., Livak, T. The eXtensible Tutor Architecture: A New Foundation for ITS. In C.K. Looi, G. McCalla, B. Bredeweg, & J. Breuker (Eds.) *Proceedings of the 12th Artificial Intelligence In Education*, Amsterdam: ISO Press (2005) 902-904

[12] Chi, M. T. H., Bassok, M., Lewis, M. W., Reimann, P., & Glaser, R. (1989). Self-explanations: How students study and use examples in learning to solve problems. Cognitive Science, 13, 145 – 182.

[13] Murray, C., VanLehn K. (2006) A Comparison of Decision-Theoretic, Fixed-Policy and Random Tutorial Action Selection. In Ikeda, Ashley & Chan (Eds.). Proceedings of the 8th International Conference on Intelligent Tutoring Systems. Springer-Verlag: Berlin. pp. 116-123. 2006.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Evaluating ACED: The Impact of Feedback and Adaptivity on Learning

Valerie J. SHUTE[1], Eric G. HANSEN and Russell G. ALMOND
Educational Testing Service, Princeton, New Jersey, USA

Abstract. This paper reports on the evaluation of a program named ACED (Adaptive Content with Evidence-based Diagnosis)—an assessment for learning system using Algebra I content related to the topic of geometric sequences. We used an evidence-centered design (ECD) approach [1] to create the system which includes three main models: proficiency, evidence, and task. Our goals of this study were to determine the learning benefit of the key design elements of adaptivity and formative feedback. Results from an experiment testing 268 heterogeneous students generally show that the system: (a) significantly improves learning, and (b) is a reliable and valid assessment tool. More specifically, the system's formative feedback feature significantly enhanced student learning, but did not detract from the accuracy of the assessment.

Keywords: Assessment for learning, Bayesian networks, evidence-centered design

1. Introduction

An assessment *for* learning (AfL) approach to education involves weaving assessments directly into the fabric of the classroom and using results as the basis to adjust instruction to promote student learning in a timely manner. This type of assessment contrasts with the more traditional, summative approach (i.e., assessment of learning), which is administered less frequently than AfL and is usually used for accountability purposes. In the past decade or so, AfL has shown great potential for harnessing the power of assessments to support learning in different content areas [2] and for diverse audiences. Unfortunately, while assessment of learning is currently well entrenched in most educational systems, assessment for learning is not.

In addition to providing teachers with evidence about how their students are learning so that they can revise instruction appropriately, AfL systems may directly involve students in the learning process, such as by providing feedback that will help students gain insight about how to improve [3], and by suggesting (or implementing) instructional adjustments based on assessment results [4]. These promises, however, need controlled evaluations to determine which features are most effective in supporting learning in a range of settings [5]. Among the AfL features that have the greatest potential for supporting student learning and which would be suitable for investigation are task-level feedback and adaptive sequencing of tasks.

1.1. Task-level Feedback

Task-level feedback appears right after a student has finished solving a problem or task, and may be contrasted with (a) general summary feedback which follows the completion of the entire assessment, and (b) specific step-level feedback which may

[1] Corresponding Author: Valerie Shute, ETS, Rosedale Rd. Princeton, NJ, USA; Email: vshute@ets.org

occur within a task [6], like ITSs often provide. Task-level feedback typically gives specific and timely (usually real-time) information to the student about a particular response to a problem or task, and may additionally take into account the student's current understanding and ability level. In general, feedback used in educational contexts is regarded as crucial to knowledge and skill acquisition (e.g., [7, 8, 9]), and may also influence motivation (e.g., [10, 11]). Immediate feedback on students' solutions to individual tasks has generally been shown to support student learning [12, 3], especially when a response or solution is wrong. That is, when a student solves a problem correctly, it usually suffices to simply provide verification of the accuracy of the response (e.g., "You are correct"). But for incorrect answers, research has suggested that it is more beneficial to provide not only verification of the incorrectness but also an explanation of how to determine the correct answer (e.g., [13, 14]). In this research, we focused on feedback for incorrect answers and evaluated the contribution to learning that elaborated feedback provides relative to simple verification feedback.

1.2. Adaptive Sequencing of Tasks

Adaptive sequencing of tasks contrasts with linear (i.e., fixed) sequencing and involves making adjustments to the sequence of tasks based on determinations such as: (a) which task would be most informative for refining an estimate of the student's proficiency level (assessment), and (b) which task would be most helpful in supporting the student's progress to a higher proficiency level (learning). Tailoring content to fit the needs of learners is quite appealing and is being incorporated into many e-learning systems, but lacks clear empirical support [5, 15]. These factors motivated the current research. Specifically, we want to test the value that adaptive task sequencing adds to learning outcome and efficiency compared to linear sequencing of tasks.

1.3. Purpose

This paper describes the evaluation of an AfL system—ACED—that combines feedback and adaptive task sequencing to support students learning Algebra I content while concurrently assessing their knowledge and skills. The goal of the evaluation was to obtain answers about whether such an AfL system works—both as an assessment tool and to support learning. Three main features of ACED were tested: feedback type, task sequencing, and proficiency estimation. The primary research questions we examine in this paper include the following: (1) Is elaborated feedback (i.e., task-level feedback that provides both verification and explanation for incorrect responses) more effective for student learning than simple feedback (verification only)? (2) Is adaptive sequencing of tasks more effective for learning than linear sequencing? (3) Does the ACED system provide reliable and valid information about the learner?

2. Methodology

For this evaluation, we used a pretest-treatment-posttest design with participating students being randomly assigned to one of four conditions. All individuals regardless of condition received two forms of a multiple choice test – one form as a pretest and the other form as a posttest. The order of forms was randomly assigned so that half the students received forms in A-B order. The control condition involved no treatment but only the pretest and posttest with an intervening one-hour period sitting at their desks

reading content that was not mathematics. Students assigned to the experimental conditions took their assigned seats at one of the 26 networked computers in the laboratory where all testing occurred. They spent the next hour solving geometric sequence problems presented on the screen.

For Research Question 1 ("Is elaborated feedback more effective for student learning than simple feedback?"), the main contrast of interest is that between Conditions 1 and 2 (holding task sequencing constant while varying type of feedback). See Table 1. Our hypothesis was that the elaborated feedback group (Condition 1) would experience greater learning than the simple feedback group (Condition 2). For Research Question 2 ("Is adaptive sequencing of tasks more effective for learning than linear sequencing?"), the main contrast of interest is that between Conditions 1 and 3 (holding feedback type constant while varying task sequencing). Our hypothesis was that the adaptive sequencing group (Condition 1) would experience greater learning than the linear sequencing group (Condition 3). While not used in a key research question, Condition 4, our Control group, serves the useful function of establishing a base level of learning resulting from no assessment-for-learning. This condition thus provides a check on the overall impact of any of the three other conditions (1, 2, and 3).

Table 1. Four Conditions Used in the Experiment

Condition	Feedback: Correct	Feedback: Incorrect	Task Sequencing
1. E/A: Elaborated feedback/ adaptive sequencing	Verification	Verification + Explanation	Adaptive
2. S/A: Simple feedback/ adaptive sequencing	Verification	Verification	Adaptive
3. E/L: Elaborated feedback/ linear sequencing	Verification	Verification + Explanation	Linear
4. Control: No assessment, no instruction	N/A	N/A	N/A

2.1. Content

The topic area of sequences was selected for implementation based on interviews with school teachers, review of state standards in mathematics, and so on. The evaluation described in this paper focuses on one branch of the proficiency model—geometric sequences (i.e., successive numbers linked by a common ratio). Shute and colleagues [16, 22] describe ACED more fully, including the proficiency model for geometric sequences, which was expressed as a Bayesian network with 8 main nodes corresponding to the key proficiencies. For each node in the proficiency model there were at least six tasks available to administer to the student—three levels of difficulty and two parallel tasks per level.

2.2. Adaptive Algorithm

ACED contained an adaptive algorithm for selecting the next item to present based on the expected weight of evidence [17, 18] provided by each item. Consider a hypothesis we want to make about a student's ability—e.g., that the student's Solve Geometric Sequences proficiency (the parent node in the model) is at or above the medium level. At any point in time, the Bayes net can calculate the probability that this hypothesis holds. Now, suppose we observe a student's outcome from attempting to solve an ACED task. Entering this evidence into the Bayes net and updating results in a change

in the probabilities. The change in log odds is called the weight of evidence for the hypothesis provided by the evidence [19]. For outcomes from tasks that have not been observed, ACED calculates the expected weight of evidence under the hypothesis [17]. The task that maximizes the expected weight of evidence (i.e., the most informative about the hypothesis) is presented next. This approach is related to the procedures commonly used in computer-adaptive testing (e.g., [20, 21]).

2.3. Authoring Tasks

The geometric sequences branch of ACED contained 63 items, each statistically linked to relevant proficiencies. A little over half of the items were multiple-choice format, with 4 to 5 options from which to choose. The remaining items required a short constructed response (a number or series of numbers). Task development entailed not only writing the items, but also writing feedback for multiple-choice answers and for common errors to constructed response items. Verification feedback was used for correct answers (e.g., "You are correct!"). For incorrect responses, the feedback provided elaboration—verification and explanation—to help the student understand how to solve the problem. To the extent feasible, the elaborated feedback for an incorrect response was "diagnostic" in the sense of being crafted based on a diagnosis of the misconception or procedural bug suggested by the student's response. If the learner entered an incorrect answer, the feedback in the simple feedback condition (S/A) would note, "Sorry, that's incorrect" and the next item would be presented. In the elaborated feedback condition (E/A), the computer would respond to the incorrect answer with the accuracy of the answer, a short and clear explanation about how to solve the problem, and often the correct answer (see [16] for specific examples of tasks, feedback, and other system details).

2.4. Sample

A total of 268 Algebra I students participated in the study. These students attended the same mid-Atlantic state suburban high school. According to their teachers, for these students, geometric sequences were not explicitly instructed as part of the curriculum, although some geometric sequence problems may have been covered as part of other topics in algebra. Our sample of students was heterogeneous in ability, representing a full range of levels of math skills: honors (n = 38), academic (n = 165), regular (n = 27), remedial (n = 30), and special education students (n = 8).

2.5. Procedure

Each two-hour session consisted of the following activities. First, all students, at their desks, received a 10-minute introduction—what they would be doing, how it would not effect their math grade, and how it was important that they try their hardest. They were also reminded that they were getting a reward for their participation. Next, all students took a 20-minute pretest. As noted earlier, we created two test forms for the pre- and posttests that were counter-balanced among all students. The tests contained 25 matched items spanning the identified content (geometric sequences) and were administered in paper and pencil format. Calculators were permitted. After the pretest, students were randomly assigned to one of four conditions where they spent the next one hour either at the computer in one of the three variants of ACED, or at their desks for the Control condition. Following the one hour period, all students returned to their desks to complete the 20-minute posttest and a paper and pencil 10-minute survey.

3. Results

3.1. Feedback and Adaptivity Effects on Learning

Research Questions 1 and 2 address the relationship to learning of (a) elaborated versus simple feedback, and (b) adaptive versus linear sequencing of tasks. Because our sample was so varied in ability level, we included two independent variables in the analysis: condition and academic level. An ANCOVA was computed with posttest score as the dependent variable, pretest score as the covariate, and Condition (1-4) and Academic Level (1-5, from honors to special education) as the independent variables. The main effects of both Condition ($F_{3,\,247} = 3.41$, $p < 0.02$) and Level ($F_{4,\,247} = 11.28$, $p < 0.01$) were significant, but their interaction was not ($F_{12,\,247} = 0.97$, NS). Figure 1 shows the main effect of condition in relation to posttest (collapsed across academic level) where the best posttest performance is demonstrated by students in the E/A condition, which also shows largest pretest-to-posttest improvement. Confidence intervals (95%) for the posttest data, per condition, are also depicted in the figure.

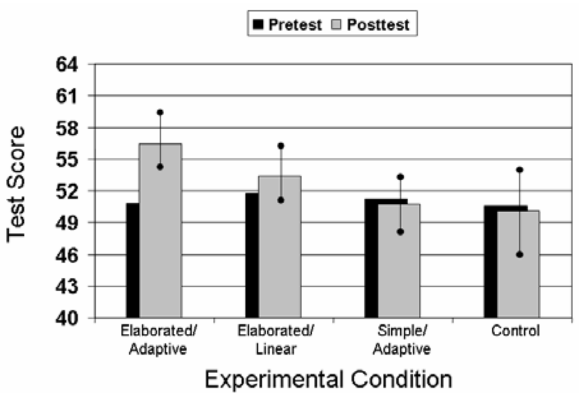


Figure 1. Condition by Pre- and Posttest Scores

The general finding of a main effect of condition on learning prompted a specific planned comparison (Bonferroni) involving the three treatment conditions in relation to posttest data. The only significant difference was between the E/A and S/A conditions, where the Mean Difference = 5.62; SE = 2.11; and $p < 0.05$. The difference between the E/A and E/L conditions (i.e., type of feedback the same but task sequencing different) was not significant. This suggests that the elaborated feedback feature was primarily responsible for the impact on learning.

3.2. Predictive Validity

The first assessment issue concerns whether the ACED proficiency estimates (relating to the 8 main nodes in the Bayes net) predict outcome performance beyond that predicted by pretest scores. Each proficiency variable possesses a triplet of probabilities, reflecting the estimated probability of being High, Medium, and Low on that proficiency. To reduce the three numbers to a single number, we calculated the Expected A Posteriori (EAP) score as: $P(\theta_{ij} = \text{High}) - P(\theta_{ij} = \text{Low})$, where θ_{ij} is the value for Student *i* on Proficiency *j*. The main outcome variable of interest for this paper is the EAP for the highest level proficiency in the model—Solve Geometric

Sequences (SGS). Higher values of EAP(SGS) should be associated with greater knowledge and skills on geometric sequence tasks. A regression analysis was computed with posttest score as the dependent variable and (a) pretest score and (b) EAP(SGS) as the independent variables. Pretest score was forced into the equation first, followed by EAP(SGS). This regression analysis provides a general sense of the validity of the ACED estimates, and whether they provided additional information beyond the pretest scores. Results showed that both of the independent variables significantly predicted student outcome data: Multiple $R = 0.71$; $F_{2, 210} = 106.57$; $p < 0.001$. Together, pretest score and EAP(SGS) accounted for 50% of the outcome variance, with EAP(SGS) accounting for 17% of the *unique* outcome variance over pretest score.

3.3. Reliability of ACED Tasks, Proficiency Estimates, and Outcome Tests

All 63 tasks in the ACED pool of geometric items were statistically linked to relevant proficiencies. Students in all three ACED conditions were required to spend one hour on the computer solving the 63 items. Thus students in the two adaptive conditions spent the same amount of time on the program as those in the linear condition. Because students in all conditions had to complete the full set of 63 items, we obtained accuracy data (scored as 0/1 for incorrect/correct) per student, per item. These performance data were analyzed using a split-half reliability test (via SPSS). The Spearman Brown split-half reliability with unequal halves (i.e., 31 and 32) was equal to 0.84, while Cronbach's $a = 0.88$ which is quite good. Next, we analyzed proficiency estimates from the Bayes net proficiency model, again using task performance data which provided input to posterior probabilities per node. These probabilities were then analyzed, making use of split-half reliabilities at the node level. The reliability of the EAP(SGS) score was 0.88. Moreover, the sub-proficiency estimates showed equally impressive reliabilities for their associated tasks suggesting they may be useful for diagnostic purposes. Separate analyses were computed on the reliabilities of the pretest and posttest items. We computed Cronbach's a for each test, then used Spearman Brown's prophecy formula to increase the size of the tests to 63 items to render the tests comparable in length to the ACED assessment. The adjusted pretest $a = 0.82$, and adjusted posttest $a = 0.85$.

3.4. Efficiency of the ACED System

In this study, we required that students in all three ACED conditions spend one hour on the computer solving all 63 items. A typical rationale for using adaptive tests relates to their efficiency capability. That is, adaptive algorithms rely on fewer items to determine proficiency level. The issue here is what the data (proficiency estimates) would look like if we required fewer items to be solved. We selected the first N (where N = 10, 15, 20, 25, 30, 40, 50, and 63) tasks from the student records, and then calculated EAP values for the parent proficiency (solve geometric sequences) from each "shortened" test. Next, we computed correlations of each of these tests with the posttest score for the students. What we expected to see was that the correlations, in general, should increase with test length until it reaches an upper asymptote related to the reliability of the posttest. We hypothesized that the data from students in the linear condition (E/L) should reach that asymptote more slowly than the data from participants in the adaptive conditions. Figure 2 shows the results of the plot, confirming our hypothesis.

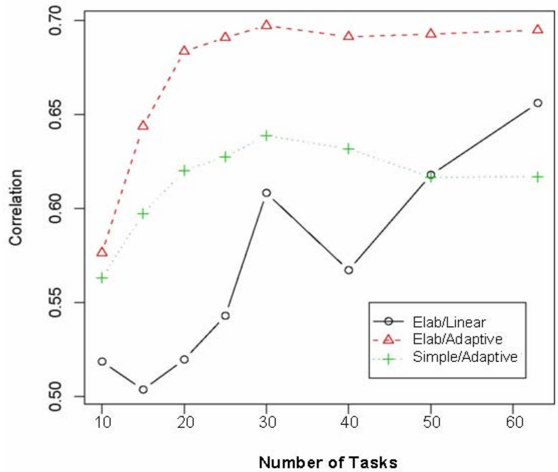


Figure 2. Correlations of EAP (SGS) with Posttest Score by ACED Condition

The quick rise and asymptote of the two adaptive conditions shows that only 20-30 tasks are needed to reach the maximum correlation with the posttest. At that juncture, for those students, the sensible next step would be instructional intervention. In the linear condition (E/L), there is a spike around task 30, then a subsequent drop. When we reviewed the list of the 63 tasks in the order in which they appeared in the linear condition, Tasks 31-36 comprised a set of very difficult items. We note that the noise associated with the correlation estimates is rather large, requiring additional investigations into the differences between correlations.

4. Conclusions

Regarding the learning question, elaborated feedback was, as hypothesized, more effective for student learning than simple feedback (verification only). Nevertheless, the study did not show adaptive sequencing of tasks to be more effective for learning than linear sequencing. However, the adaptive condition did show greater efficiency than the linear condition in achieving high reliability and validity. For students in the adaptive condition, the ACED assessment could have reasonably terminated after approximately 20 items with no degradation in prediction of outcome. Administering a 20-30 minute test (students averaged 1 minute per item) should yield valid and reliable results efficiently—for students and teachers.

The ACED system was very reliable in relation to the ACED tasks, proficiency estimates, and pretests and posttests. Based on our findings, using evidence-based assessment design with Bayes net technology facilitates valid estimation of proficiencies from performance data. Regression analysis showed just a single proficiency estimate—EAP(SGS)—can significantly predict posttest performance, beyond that provided by pretest performance. In conclusion, we envision a role for the ACED algorithm in a larger instructional support system. The adaptive AfL system would continue to accurately assess the student until the best instructional options for that student become clear. Meanwhile, elaborated feedback would ensure that the assessment itself was a valid learning experience.

References

[1] Mislevy, R. J., Steinberg, L. S., & Almond, R. G. (2003). On the structure of educational assessments. *Measurement: Interdisciplinary Research and Perspectives, 1*, 3–62.

[2] Black, P., & Wiliam, D. (1998). Assessment and classroom learning. Assessment in Education: Principles, *Policy & Practice, 5*(1), 7-74.

[3] Shute, V. J. (2007). *Focus on formative feedback*. ETS Research Report, Princeton, NJ.

[4] Stiggins, R. J. (2002). Assessment Crisis: The Absence of Assessment FOR Learning, *Phi Delta Kappan Professional Journal, 83*(10), 758-765.

[5] Jameson, A. (2006). Adaptive interfaces and agents. In J. A. Jacko & A. Sears (Eds.), *Human-computer interaction handbook* (2nd ed.). Mahwah, NJ: Lawrence Erlbaum.

[6] VanLehn, K. (2006). The behavior of tutoring systems. *International Journal of Artificial Intelligence in Education, 16*(3), 227-265.

[7] Azevedo, R., & Bernard, R. M. (1995). A meta-analysis of the effects of feedback in computer-based instruction. *Journal of Educational Computing Research, 13*(2), 111-127.

[8] Bangert-Drowns, R. L., Kulik, C. C., Kulik, J. A., & Morgan, M. T. (1991). The Instructional Effect of Feedback in Test-like Events. *Review of Educational Research, 61*(2), 213-238.

[9] Moreno, R. (2004). Decreasing cognitive load for novice students: Effects of explanatory versus corrective feedback in discovery-based multimedia. *Instructional Science, 32,* 99-113.

[10] Lepper, M. R. & Chabay, R. W. (1985). Intrinsic motivation and instruction: Conflicting views on the role of motivational processes in computer-based education. *Educational Psychologist, 20*(4), 217-230.

[11] Narciss, S. & Huth, K. (2004). How to design informative tutoring feedback for multi-media learning. In H. M. Niegemann, D. Leutner & R. Brunken (Ed.), *Instructional design for multimedia learning* (pp. 181-195). Munster, New York: Waxmann.

[12] Corbett, A. T., & Anderson, J. R. (2001). Locus of Feedback Control in Computer-based Tutoring: Impact on Learning Rate, Achievement and Attitudes. *Proceedings of ACM CHI 2001 Conference on Human Factors in Computing Systems.* New York, ACM Press: 245-252.

[13] Kluger, A. N., & DeNisi, A. (1996). The effects of feedback interventions on performance: A historical review, a meta-analysis, and a preliminary feedback intervention theory. *Psychological Bulletin, 119*(2), 254-284.

[14] Mason, B. J., & Bruning, R. (2001). *Providing feedback in computer-based instruction: What the research tells us*. Center for Instructional Innovation, University of Nebraska-Lincoln: 14. Retrieved November 1, 2006, from http://dwb.unl.edu/Edit/MB/MasonBruning.html

[15] Shute, V. J. & Zapata-Rivera, D. (in press). Adaptive technologies. In J. M. Spector, D. Merrill, J. van Merriënboer, & M. Driscoll (Eds.), *Handbook of Research on Educational Communications and Technology* (3rd Edition). Mahwah, NJ: Erlbaum Associates.

[16] Shute, V. J., Hansen, E. G., & Almond, R. G. (in press). *An assessment for learning system called ACED: Designing for learning effectiveness and accessibility,* ETS Research Report, Princeton, NJ.

[17] Good, I. J., & Card, W. (1971). The diagnostic process with special reference to errors. *Method of Inferential Medicine, 10*, 176–188.

[18] Madigan, D., & Almond, R. G. (1996). On test selection strategies for belief networks. In D. Fisher & H.-J. Lenz (Eds.), *Learning from data: AI and statistics IV* (pp. 89–98). New York: Springer-Verlag.

[19] Good, I. J. (1985): Weight of evidence: A brief survey. In J. Bernardo, M. DeGroot, D. Lindley, & A. Smith (Eds.) *Bayesian Statistics 2* (pp. 249-269), Amsterdam: North Holland.

[20] Conejo, R., Guzman, E., Millán, E., Trella, M., Perez-De-La-Cruz, J. L., & Rios, A. (2004). SIETTE: A web-based tool for adaptive testing. *International Journal of Artificial Intelligence in Education 14*(1), 29-61.

[21] Wainer, H., Dorans, N. J., Flaugher, R., Green, B. F., Mislevy, R. J., Steinberg, L., & Thissen, D. (2000). *Computerized adaptive testing: A primer* (2nd Edition), Mahwah, NJ: Erlbaum Associates.

[22] Shute, V. J., Graf, E. A., & Hansen, E. G. (2006). *Designing adaptive, diagnostic math assessments for individuals with and without visual disabilities*. ETS Research Report, RR-06-01, Princeton, NJ.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Modeling learning patterns of students with a tutoring system using Hidden Markov Models

Carole BEAL [a,1], Sinjini MITRA [a] and Paul R. COHEN [a]
[a] *USC Information Sciences Institute*

Abstract. The current paper focuses on modeling actions of high school students with a mathematics tutoring system with Hidden Markov Models. The results indicated that including a hidden state estimate of learner engagement increased the accuracy and predictive power of the models, both within and across tutoring sessions. Groups of students with distinct engagement trajectories were identified, and findings were replicated in two independent samples. These results suggest that modeling learner engagement may help to increase the effectiveness of intelligent tutoring systems since it was observed that engagement trajectories were not predicted by prior math achievement of students.

Keywords. clustering, Hidden Markov Model, inference, intelligent tutoring system, learning action patterns, learner engagement, prediction, transition matrices

1. Introduction

Intelligent tutoring systems (ITS) research is one of the "success stories" of artificial intelligence: A number of studies have demonstrated that a detailed model of the learner's domain knowledge can be used to provide individualized instruction, and to accelerate student learning well above what would be expected in a whole-class context (Corbett & Anderson, 1995; Koedinger et al., 2000). Traditionally, tutoring systems researchers have focused primarily on modeling the learner's cognitive processes while solving problems, i.e., the "model tracing" approach. However, there is growing recognition that learning involves more than cognition, and that students' actions with an ITS also reflect "engagement", meaning the transient shifts in attention and the emotions that are often associated with learning (Qu & Johnson, 2005; Vicente & Pain, 2002). For example, a student may become bored or fatigued over the course of a session and deliberately enter incorrect answers in order to elicit the correct answer from the ITS (Baker et al., 2005).

Including estimates of engagement in learner models could be used to improve the effectiveness of tutoring systems, for example, by shifting the type of problem being delivered if boredom is detected, or by providing supportive feedback if the student seems frustrated. However, relatively little is known about modeling learners' engagement while using an ITS. One approach is to use sensors to capture physiological indices

[1]Correspondence to: Carole Beal, USC Information Sciences Institute, 4676 Admiralty Way, Suite 1001, Marina del Rey, CA 92647. Tel.: +1 310 448 8755; Fax: +1 310 822 0751; E-mail: cbeal@isi.edu.

of emotional states such as interest, boredom, and frustration (D'Mello et al., 2004). Although very promising in the laboratory, this approach is difficult to scale for use by large numbers of students in real classroom situations in which intrusive and fragile equipment cannot easily be used. Here, we investigate an alternative approach: model students' action traces using a hidden variable denoting their engagement level as they work with an ITS (Beck, 2005).

The rest of the paper is organized as follows. Section 2 describes the data that we use for our experiments. The statistical model is introduced in Section 3, which also includes results and analysis. Conclusions and future work are presented in Section 4.

2. Data Sources

Our analyses focused on records of student actions with a tutoring system for high school mathematics. Data from two independent samples were available: the "CA" sample included 91 students who completed an average of 20 problems, and the "BA" sample included 122 students who completed an average of 30 math problems. The BA sample was more heavily weighted towards lower-achieving students, whereas the CA sample included a broader range of achievement levels. CA students attempted a maximum of 5 sessions and BA students a maximum of 3 sessions.

In both samples, the data were automatically logged as students worked with the mathematics tutoring system during their regular school instruction. Students worked on a series of math problems, each including a figure, graph or table; the problem text and equation; and five answer options. On each problem, the student could choose answers and receive feedback on accuracy (e.g., a click on an incorrect answer elicited a red "X" whereas the correct answer elicited a green checkmark); request a series of multimedia hints leading to the solution; or move on to the next problem without answering.

Prior work had shown that a student's actions associated with a math problem could be machine-classified into five categories: *Guessing/Help Abuse*; *Independent-Accurate problem solving*; *Independent-Inaccurate problem solving*; *Learning with Help*; and *Skipping*, with 95% accuracy (Beal et al., 2006). In the present study, the action and latency data recorded on each math problem were similarly machine classified into 5 action patterns. Thus, each student's data record included an ordered sequence of action patterns representing her or his behavior on the math problems over the course of the tutoring session. An example of a student's data is shown in Table 1. The corresponding action sequence is [AAECBD...].

Table 1. Sample action pattern sequence.

Problems	1	2	3	4	5	6	...
Action Patterns	Guess	Guess	Skip	Ind-accurate	Learn	Ind-inaccurate	...
Code	A	A	E	C	B	D	...

3. Statistical Model and Results

We model the performance of students with *Hidden Markov Models* (HMM). Although HMMs have been applied extensively in speech processing (Rabiner, 1989) and video-

based face identification (Liu & Chen, 2003), the application of HMMs to tutoring systems is relatively new, particularly with regard to modeling the behavior of individual students rather than groups (Stevens et al., 2005). We fit HMMs to the sequence of actions emitted by individual students, with 3 hypothesized hidden states corresponding to a student's "engagement level" $-$ *low, medium* and *high*, that we assume influences student interactions with the ITS. Having fit an HMM to students' action pattern data, the transition matrices help us determine their probabilities of moving from one level of engagement to another, as well as of remaining in a particular engagement state while attempting a series of problems. In other words, if a student's engagement level is known at time t, the transition matrix will tell us his/her likely engagement level at time $t+1$.

The average transition matrices over all students in each sample from fitting HMMs to the actions patterns of students in Session 1 only are shown in Table 2. The rows represent time t, and the columns represent time $t+1$, so that a particular entry represents the probability of transitioning from a particular engagement level at time t to another one at time $t+1$. For example, CA students who are in a low engagement state have a 45.63% chance of moving into a high engagement state at the next time step. The diagonal entries represent the probabilities of remaining in the same engagement level from one time step to the next. For example, a CA student has a 30.14% chance of remaining in the low engagement state at time $t+1$ if he is in such a state at time t.

Table 2. Average transition matrices for CA and the BA samples.

Sample	Hidden state	Low	Medium	High
CA	Low	0.3014	0.2423	0.4563
	Medium	0.1982	0.4721	0.4176
	High	0.1517	0.3312	0.6050
BA	Low	0.4727	0.1966	0.3307
	Medium	0.1720	0.5583	0.2696
	High	0.1470	0.2772	0.5758

Both transition matrices suggest that there is a degree of "inertia" in students' actions, i.e., the probabilities for persisting in the same state are generally higher than those of shifting to another state. For example, in both samples, the highest probabilities are observed for students who are likely to persist in that state. However, there is considerable individual variation in the engagement trajectories (not shown due to space constraints). We address this issue in the next section.

3.1. Clustering Students based on HMM

In order to account for the variance in the individual transition matrices, we clustered students in each dataset based on the Kullback-Leibler (KL) distance between the individual transition matrices (Ramoni, Sebastiani & Cohen, 2001). Such groups represent students who follow similar trajectories through the different levels of engagement during an ITS session. We obtained 3 groups for the CA sample, and 4 groups for the BA sample. The average transition matrix for each group appear in Tables 3 and 4 respectively.

We see that there was considerable similarity in the clustered HMM groups across the two samples. More specifically, both samples include one fairly large cluster of students who tended to persist in medium and high engagement states ($50-65\%$) ("steady

Table 3. Group transition matrices for 91 CA students. N denotes the number in each group.

Group	Hidden state	Low	Medium	High
1	Low	0.3641	0.1008	0.5351
N=51	Medium	0.2105	0.5801	0.2094
56%	High	0.1957	0.2928	0.5115
2	Low	0.2606	0.4899	0.2495
N=34	Medium	0.0384	0.3933	0.8036
37%	High	0.1124	0.4086	0.7143
3	Low	0.0000	0.0417	0.9583
N=6	Medium	1.0000	0.0000	0.0000
7%	High	0.0000	0.2189	0.7811

Table 4. Group transition matrices for 122 BA students. N denotes the number in each group.

Group	Hidden state	Low	Medium	High
1	Low	0.5056	0.1089	0.3855
N=81	Medium	0.1392	0.6637	0.1971
66%	High	0.0970	0.2201	0.6830
2	Low	0.2864	0.6809	0.0327
N=20	Medium	0.0260	0.2839	0.6901
17%	High	0.2970	0.1455	0.5574
3	Low	0.1735	0.1429	0.6837
N=7	Medium	0.9429	0.0000	0.0571
6%	High	0.0000	0.1805	0.8105
4	Low	0.7105	0.0344	0.2551
N=14	Medium	0.1648	0.6152	0.2201
11%	High	0.3064	0.6936	0.0000

state"). The probability of moving to the low engagement state is only about 20% or less for these students. A second group with a distinctly different profile is also observed in both samples: these students tend to become more engaged over time ("increasing engagement"), e.g., from medium to high engagement state (80% for CA, 69% for BA), and are not likely to move to a low engagement state. Both samples also include a small third group characterized by strong shifts in engagement ("fluctuating engagement"). One shift is from low to high engagement (95% for CA, 68% for BA), and another from medium to low engagement (100% for CA, 94% for BA). Finally, the BA sample includes a fourth group ("declining engagement") characterized by persisting in a low engagement state (71%) and moving from high to medium state (69%). Recall that the BA sample included more students with low achievement.

3.1.1. Behavior of students in HMM groups

In order to compare the behavior of students in the HMM clusters, we looked at the mean proportion scores for the 4 primary action patterns (rates for "Problem Skipping" were very low and are not analyzed here). The results were again fairly consistent across the two samples. For the CA sample, an analysis of variance (ANOVA) with HMM group as the grouping factor and proportion of actions classified as Guessing as the dependent measure showed a main effect of HMM Group, $F(2,97) = 4.346$, $p < .05$, which

implied that the amount of guessing varied significantly among the different groups. Tukey's pairwise comparison tests further indicated that Groups 1 and 3 had a significantly higher proportion of Guessing than Group 2. For the BA sample we observed $F(3, 109) = 9.436,\ p < .01$. Group 4 has significantly high Guessing scores than the other three groups and Groups 2 and 3 had significantly lower scores than the other two.

With regard to the effective use of multimedia help (Learn), there were no significant differences between the HMM clusters for the CA sample. For the BA sample, ANOVA showed that the 4 groups of BA students varied considerably with respect to the amount of learning ($F(3, 109) = 4.968,\ p < .01$). Tukey's tests indicated that the means for Groups 1, 2, 3 were similar to each other, but significantly higher than the Group 4 mean.

With regard to Independent-Inaccurate problem solving, there was a main effect of HMM Group for the CA sample, $F(2, 97) = 3.784, p < .05$ (although Tukey's test did not reveal any significant differences in the group means), but none for the BA sample.

ANOVA conducted on the proportion scores for Independent-Accurate problem solving showed no difference for the HMM groups for either sample. However, as shown in Table 5, mean scores for the CA sample were generally higher than for the BA sample, consistent with the fact that the BA sample included more low-achieving students.

Table 5. Mean Proportion scores for the HMM groups in the 2 samples

HMM Group	CA1	CA2	CA3	BA1	BA2	BA3	BA4
Guess	0.21	0.10	0.19	0.22	0.08	0.09	0.46
Learn	0.20	0.23	0.30	0.25	0.39	0.42	0.13
Ind. Accurate	0.27	0.39	0.39	0.19	0.20	0.22	0.10
Ind. Inaccurate	0.24	0.18	0.09	0.20	0.21	0.16	0.18

3.1.2. HMM clusters and math achievement

It may be possible that the HMM groups simply capture aspects of student engagement with the ITS which can be directly traced to their mathematics proficiency. For example, students who do not have very good math skills might be more likely to view the multimedia help features; students with strong math ability might be more likely to solve problems accurately and independently. This plausible interpretation would be consistent with the model-tracing view of student learning, i.e., that students' actions result from their cognitive understanding of the domain. However, in the present study we did not find a relation between students' math achievement and their HMM group membership, as explained below.

In the CA sample, the students' classroom teachers had provided independent ratings of each student's math proficiency before the start of the study, based on their knowledge of the students' performance on homework, tests and exams. A chi-square analysis indicated that there was no significant association between the students' HMM cluster membership and their ratings on math achievement. In the BA sample, students had completed a 42-item pre-test of math skills before starting to work with the ITS. A logistic regression analysis showed that there was no association between students' pre-test scores and HMM group membership. This suggests that math proficiency does not completely explain students' behavior with the ITS, as observed in two independent samples with different types of measurement. This is consistent with the view that ITS student models may be enhanced with estimates of engagement as well as domain knowledge.

3.2. Prediction based on HMM

In this section, we describe additional work designed to predict a student's likely engagement trajectory, with the goal of diagnosing the student's behavior early enough that interventions could be deployed to increase or sustain engagement. First of all, we used the fitted HMM for each individual student estimated from the action patterns on the first 15 problems in Session 1 to predict the action pattern at the next step for problems $M = 16, \ldots, S$ (where S is the number of problems attempted by each student). The results in Table 6 show that the prediction accuracy was 42.13% for the CA sample, and 48.35% for the BA sample. This is well above chance or random guessing (25%), which denotes that a student is likely to take any of the 4 primary actions at each time (i.e., assuming no prior knowledge about the system). This indicates that HMMs can be useful in predicting students' future behavior with the tutoring system beyond mere random guessing. Next we used the group HMMs to perform prediction by using the group transition matrix and group emission matrix corresponding to each student. Not surprisingly, the group HMM accuracy was poorer than the individual HMMs, yet they were more accurate than chance.

On comparing the prediction accuracies with those obtained from simple Markov Chain (MC) models which do not assume the existence of a hidden state, we find that the individual MCs performed less well than the individual HMMs, although still better than chance. Moreover, the group HMM models performed better than the individual MCs for the BA sample and did not differ significantly for the CA sample. Noticeably, the predictive power of the group MCs is the poorest. Thus, we conclude that the HMMs with the hidden state can model the variability in students' behavior during a problem-solving session more effectively than the MCs.

Table 6. Prediction accuracies for HMMs and MCs on Session 1.

	Ind. HMM	Ind. MC	Gr. HMM	Gr. MC
CA	42.13%	33.43%	34.40%	18.73%
BA	48.35%	32.48%	26.00%	15.52%

3.2.1. Predicting Future Sessions

The next set of experiments are based on students who have two sessions with the ITS – 10 CA students and 25 BA students. Here, for each student, we train the HMM based on all the problems he or she attempts in Session 1. We then use the second sessions of these students for validating our models using the fitted HMMs (both individual and group models) from the first sessions. We generate predictions for all the problems attempted in Session 2 for each of these students. We again compare our prediction results from the HMMs to those from MC models that do not take into account any hidden variable information. Table 7 shows these prediction accuracies. The highest accuracies for multiple sessions are obtained with the individual HMMs, followed by the group HMMs. Both are significantly better than pure random guessing or chance (25% for 4 possible actions), whereas the accuracies of the MCs are worse than chance. Note that, predictive abilities on future session are lower than that on the same session, as expected. This is because often the dynamics of student behavior are more stable over a session than across sessions.

Table 7. Prediction accuracies for Session 2.

	Ind. HMM	Ind. MC	Group HMM
CA	45.93%	10.15%	30.12%
BA	36.50%	12.53%	25.22%

3.2.2. *Qualitative Assessment of Prediction Accuracies*

The above results indicate that HMMs fitted to students' actions in one session will predict their actions in subsequent sessions with about 40% accuracy. This is certainly not perfect, but it is well above knowing nothing at all about the student in terms of designing interventions to sustain engagement, especially if we could distinguish those students for whom we could predict engagement on future sessions from those where our predictions are not likely to be accurate. Therefore, we next investigated if the prediction accuracy on the first session could signal the prediction accuracy on a future session using the individual HMMs that had the best accuracies. Both samples show significant positive correlations (0.8851 for CA and 0.7984 for BA) between the prediction accuracies from the 2 sessions. Thus, we conclude that if the HMMs are good at predicting the first session, they are likely to have high prediction accuracies on subsequent sessions as well. On the other hand, if an individual's HMM is not good at predicting the first session, it is unlikely to be very useful for predicting engagement trajectory in future sessions.

Comparing these individual prediction accuracies with the distributions of the action patterns on the two sessions, we find that students with similar prediction accuracies for both sessions typically have low entropy between the two distributions (measured by KL distance), and vice versa for students who have poor accuracies on both sessions. There are, however, a few students with low accuracies on Session 2 despite having medium accuracies on Session 1. This happens because these students miss certain action patterns on Session 1 that appear frequently on Session 2, thus leading to poor predictions.

4. Conclusions and Future Work

Our primary goal in this study was to use a student's actions with a tutoring system to estimate his or her engagement: an unobservable state that is hypothesized to influence the observed actions emitted by the student. Hidden Markov Models are an ideal analytic approach because these models allow for explicitly modeling unobservable influences. Here, we fit HMMs to students' actions with a math tutoring system, with a three-level hidden state corresponding to "engagement". The results indicated that students' action sequences could be modeled well with HMMs, that their actions at one point in time could be successfully used to predict the subsequent action, and that the prediction accuracy of these models were superior to simple Markov Chain models that did not include a hidden state. We also found that, in many cases, the HMM models from Session 1 could be used to predict students' future engagement trajectories on Session 2.

By clustering the HMMs, we also identified groups of students with distinct trajectories through the hidden state space. Similar groups were observed in two independent samples from studies conducted in different school systems and geographical locations, suggesting that the patterns may be fairly general and consistent. The largest group included students who were generally engaged with the tutoring system, and whose engagement tended to remain stable over the course of the session. A second group was also

quite engaged but showed a stronger tendency to become more highly engaged over time. One interpretation is that students in these groups do not really require any intervention from the ITS to support their learning. Thus this facilitates evaluation of the impact of interventions by looking for shifts in a student's HMM cluster, and hence design more effective ITS. As a next step, we will use more refined models, such as non-stationary HMMs to explicitly model the amount of time a student spends on a problem.

The primary limitation of the study is that we do not have direct evidence that the hidden state in the HMMs actually corresponds to any of the processes that are known to reflect "engagement" in the learner, including transient shifts in the learner's attention, emotions and cognitive effort. But we still chose to term the hidden state "engagement" because we believe it to influence a student's behavior with a math tutoring session the most. However, regardless of the true nature of the hidden state, HMMs have proved to be an efficient modeling tool for students' action patterns.

Acknowledgment: The research was supported by the National Science Foundation (REC 0411886). The views expressed are not necessarily those of the funding agency. We would also like to thank the participating schools and project members Erin Shaw and Lei Qu.

References

[1] R.S. Baker, A.T. Corbett, K.R. Koedinger and I. Roll (2005). Detecting when students game the system, across tutor subjects and classroom cohorts, In L. Ardissono, P. Brna, & A. Mitrovic (Eds.), *User Modeling*, 220-224. Berlin: Springer-Verlag.

[2] C.R. Beal, L. Qu and H. Lee (2006). Classifying learner engagement through integration of multiple data sources. *Proceedings of AAAI'06*, Boston, MA.

[3] J.E. Beck (2005). Engagement tracing: Using response times to model student disengagement. In C-K. Looi et al. (Eds.), *Artificial intelligence in education: Supporting learning through intelligent and socially informed technology*, 88-95. Amsterdam: IOS Press.

[4] A.T. Corbett and J.R. Anderson (1995). Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User Adapted Interaction*, **4**: 253-278.

[5] S.K. D'Mello, S.D. Craig, B. Gholson, S. Franklin, R. Picard and A.C. Graesser (2004). Integrating affect sensors into an intelligent tutoring system. In *Affective Interactions: The Computer in the Affective Loop. Proceedings of the 2005 International Conference on Intelligent User Interfaces*. 7-13. New York.

[6] K.R. Koedinger, A.T. Corbett, S. Ritter and L.J. Shapiro (2000). Carnegie Learning's Cognitive Tutor: Summary of research results. *White Paper, Carnegie Learning*. Pittsburgh PA.

[7] X. Liu and T. Chen (2003). Video-based face recognition using adaptive Hidden Markov Models. *Proceedings of CVPR'03*, Madison, Wisconsin.

[8] L.R. Rabiner (1989). A tutorial on HMM and selected applications in speech recognition. *In Proceedings of IEEE*, **77**(2):257-286.

[9] L. Qu and W.L. Johnson (2005). Detecting the learner's motivational states in an interactive learning environment. In C-K. Looi et al. (Eds.), *Artificial intelligence in education: Supporting learning through intelligent and socially informed technology*, 547-554. IOS Press.

[10] M. Ramoni, P. Sebastiani and P.R. Cohen (2001). Bayesian clustering by dynamics. *Machine Learning*, **47**:91-121.

[11] R. Stevens, D. Johnson and A. Soller (2005). Probabilities and prediction: Modeling the development of scientific problem solving skills. *Cell Biology Education*, **4**:42-57. Springer.

[12] A. Vicente and H. Pain (2002). Informing the detection of the student's motivational state: An empirical study. *In 6th International Conference on Intelligent Tutoring Systems*, 933-943. Biarritz, France.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Exploring Alternative Methods for Error Attribution in Learning Curves Analysis in Intelligent Tutoring Systems

Adaeze NWAIGWE[1], Kenneth R. KOEDINGER[1], Kurt VANLEHN[2], Robert HAUSMANN[2] and Anders WEINSTEIN[2]
[1]*Human-Computer Interaction Institute, Carnegie Mellon University*
5000 Forbes Avenue, Pittsburgh, PA 15213, USA
anwaigwe@cs.cmu.edu, koedinger@cmu.edu
[2]*LRDC, University of Pittsburgh, PA, USA*
[vanlehn, bobhaus, andersw,]@pitt.edu

Abstract. It is becoming a standard technique to use learning curves as part of evaluation of intelligent tutoring systems [1,2,3], but such learning curves require a method for attributing errors. That is, the method must determine for each error a student makes what "knowledge component" in the student model is to blame. To this point, alternative methods for error attribution have not been systematically investigated. We implemented four alternative methods for error attribution – two temporal heuristics and two location-temporal heuristics. We employed two evaluation standards – a statistical standard for measuring model fit and parsimony and the Kappa technique for measuring inter-observer reliability. We looked to see which method better met the "learning-curve standard" that is, led to better prediction of students' changes in error rate over time. Second, we asked if the codes generated by the methods better met the "human-match standard", that is, were they like error attributions made by human coders. Both evaluation standards led to better results for the location-temporal heuristic methods than the temporal heuristic methods. Interestingly, we found that two of the methods proposed were better at error attribution, according to the learning-curve standard, than the original cognitive model of the intelligent tutoring system. Overall, these results suggest that the heuristics proposed and implemented in this paper can generally aid learning curve analysis and perhaps, more generally, the design of student models

Introduction

Increasingly, learning curves have become a standard tool for measurement of students' learning in intelligent tutoring systems. Slope and fit of learning curves show the rate at which a student learns over time, and reveal how well the system model fits what the student is learning. However, these learning curves require a method for attributing error to the "knowledge components" (skills or concepts) in the student model that the student is missing.

In order to facilitate learning curve analysis which can also be referred to as internal evaluation [11], we have proposed and implemented four automated heuristics for attributing error to the desired knowledge components, that is, the skills which the student is required to acquire, during the course of problem solving. These are two temporal heuristic models and two temporal-location heuristic hybrid models. We specifically addressed the following research questions:

1. Which heuristic better predicts students' changes in error rate over time?
2. Which heuristic produces error attribution codes that match those of human coders, best?

To address question 1, we applied the statistical component of learning factors analysis [4], while in answering question 2, Kappa analysis was used to compare codes from each heuristic to those produced by human coders.

Section 1 is a brief review of literature, section 2 describes the data source used in our analysis, while section 3 illustrates our methodology. Results, discussions and conclusions are presented in sections 4, 5 and 6, respectively.

1. Literature Review

The power relationship between the error rate of performance and the amount of practice is shown in equation 1 [5]. The relationship shows that the error rate decreases as the amount of practice increases. As previously mentioned, this function is known as a "learning curve".

$$Y = aX^b \quad \text{........(1)}$$

where
Y = the error rate
X = the number of opportunities to practice a knowledge component (e.g., skill, concept, rule, constraint)
a = the error rate on the first trial, reflecting the intrinsic difficulty of a knowledge component
b = the learning rate, reflecting how easy a knowledge component is to learn

A good cognitive model is expected to follow the power law. A cognitive model is a set of production rules or skills encoded in intelligent tutors to model how students solve problems. Productions embody the knowledge that students are trying to acquire, and enable the tutor estimate each student's learning of specific skills as the student works through exercises [12].

Cen and colleagues proposed a semi-automated method for improving a cognitive model called Learning Factors Analysis (LFA) [4]. LFA is a three-component system, implemented in Java, which combines statistics, human expertise and combinatorial search to evaluate and improve a cognitive model. The statistical component quantifies the skills and provides information on data fit – this is the component we make use of in this study.

While the power law model applies to individual skills, it does not include student effects. In order to accommodate student effects for a cognitive model that has multiple rules, and that contains multiple students, Cen et al extended the power law model to a logistic regression model (equation 2 below) based on the following assumptions:

1. Different students may initially know more or less. An *intercept* parameter for each student reflects this.
2. Students learn at the same rate. Thus, *slope* parameters do not depend on student. This simplification is to reduce the number of parameters in equation 2 and also because the focus is on refining the cognitive model rather than evaluating student knowledge growth [6,7].
3. Some knowledge components are better known than others. An intercept parameter for each knowledge component captures this.
4. Some skills are easier to learn than others. Thus, this is reflected using a slope parameter for each skill

$$\ln[p/(1-p)] = B_0 + \Sigma\, \alpha_j X_i + \Sigma\, \beta_j Y_j + \Sigma\, \gamma Y_j T_j \quad \text{.......(2)}$$

where
p = the probability of success at a step performed by student i that requires knowledge component j
X = the dummy variable vector for students; Y = the dummy variable vector for knowledge components; T = the number of practice opportunities student i has had on knowledge component j; α = the coefficient for each student, that is, the student intercept; β = the coefficient for each knowledge component, that is, the knowledge component intercept
γ = the coefficient for the interaction between a knowledge component and its opportunities, that is, the learning curve slope

In this paper, we use the statistical component of LFA to quantify the skills and evaluate model fit and parsimony with respect to equation 2. Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC) are two estimators for prediction risk [10] used in LFA and their formulae are shown in equations 3 & 4. Lower AIC & BIC scores, mean a better balance between model fit and complexity. BIC however, imposes a more severe penalty for complexity than AIC.

$$AIC = -2*\text{log-likelihood} + 2*K \text{............. (3)}$$

$$BIC = -2*\text{log-likelihood} + K * \ln(n) \text{.......... (4)}$$

where
log-likelihood measures model fit,
K = number of covariates in equation 2, and measures model complexity,
n = number of observations

2. Data Source

To test the methods, we used data from the Andes physics tutor [8] collected at the US Naval Academy during its regular physics class (see figure 1). This data was collected as part of the Pittsburgh Science of Learning Center's LearnLab facility that provides researchers, access to run experiments in or perform secondary analyzes of data collected from one of seven available technology-enhanced courses running at multiple high school and college sites (see http://learnlab.org). The data spanned 4 multi-step physics problems solved by 104 students and involved about 18,000 observed "transactions".

In Andes, the student is required to define all variables before entering them in an equation. One method for defining variables is to draw a vector. In figure 1, the student correctly defines the vector, 'F_W', using a drawing (trn 1). The student also successfully defines the mass of the car and its travel distance (trns 2 & 3). However, in trn 4, the student writes a force equation, using the variable 'g' which the student is yet to define. This transaction is incorrect. Usually, Andes would indicate success at completing a transaction by turning the entry green. Entries for incorrect transactions are turned red.

The general problem of error attribution is to determine for each incorrect or hint transaction (solicited or unsolicited help from the ITS) what knowledge component in the overlay student model, is to blame. One common solution is that the intelligent tutoring system provides this information (as was sometimes the case in our data set). However, in some log data this information is not available and, furthermore, the error attribution method used by the ITS might not always be the best choice. An alternative strategy is to search in the log for a subsequent correct student action and attribute the error to the knowledge component coding of that correct action. If that subsequent correct action is the next one in time, then one is employing the "temporal heuristic" described below. However, sometimes students jump around in a tutor interface and the next correct action may not be related to the error – it might be better to search for the next correct action that is in the same interface location in which the error occurred. This "location heuristic" has been employed in past approaches [1, 2, 3] and is further described below.

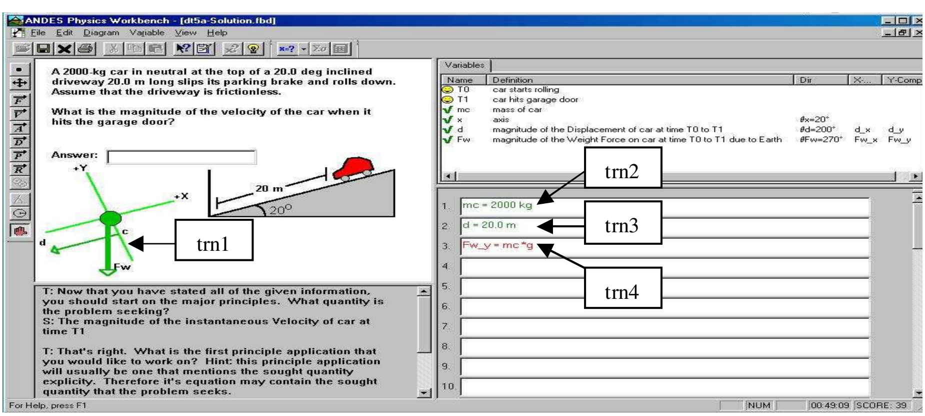


Figure 1. Example Screenshot of the Andes Intelligent Tutor

3. Methodology

The four heuristics proposed are the two temporal heuristics – TH_1, and TH_2 and two location-temporal heuristics LH_1 and LH_2. Each of heuristics is implemented in pure Java 1.5 and is described below.

3.1. Temporal Heuristic (TH) Methods

In seeking a knowledge component (KC) code for an error transaction for which the KC code is missing, the two TH methods generally assign blame to the first KC the student successfully implements. While the TH_2 method is implemented on all "incorrect" and "hint" transactions, that is, all error transactions, TH_1 is implemented on only error transactions for which the Andes tutor failed to ascribe blame to at least one or more KCs. When Andes performs blame attribution for an error transaction, it usually ascribes blame to the KC or set of KCs, in the problem's solution set, which it believes is responsible for the student's error.

To illustrate how TH_1 is implemented, the java program reads the first student/tutor transaction in an XML log file. If it is an error transaction which has a missing KC code, the program searches down the log till it finds the first correct transaction for the same problem. The KC code for this later transaction is then assigned as KC code to the afore-mentioned error transaction for which a code is sought. If no correct transaction is found, the TH method ends for that error transaction without returning a code. The TH sequence is implemented on all error transactions till the end of the log file is reached. Error transactions for which no KC code is found are excluded from the analysis.

3.2. The Location Heuristic (LH) Method

Let us assume a student makes an error at a certain interface location and a KC code to blame, is sought. When it can, the LH method uses the error location as a heuristic for deciding how to assign blame to the first KC, which the student successfully implements at that same location. While LH_2 implements this method on all error transactions, LH_1 implements the method on only error transactions for which Andes failed to ascribe blame to a KC. The step for the LH method is given in section 3.2.1. With respect to LH_1, the java program reads the first student/tutor transaction in an XML log file. If it is an error transaction which has a missing KC code, the program searches down the log till it finds the first correct transaction, implemented at the same location as the error transaction and for the same problem. The KC code for the correct transaction becomes the KC code for the error transaction for which a code is sought. If no correct transaction is found for the same location as the error transaction, the LH method ends for that error transaction and repeats its sequence again on the next error transaction till it reaches the end of the log file. Again, error transactions for which no KC code is found are excluded from the analysis.

3.2.1. Algorithm for the Location-Temporal Heuristic

1. Read student/tutor transaction. If error transaction, go to 2. Else, go to 3.
2. Does KC code exist? If yes, go to 3. If no, go to 4

3. Last transaction? If yes, STOP. If no, go to 1
4. Does record of actual interface location exist? if yes, go to 5. if no, go to 6
5. Find 1st correctly implemented KC at same location as error transaction. Acquire KC code. If successful, go to 3. If unsuccessful, go to 7
6. Find 1st correctly implemented KC. Acquire KC code. If successful, go to 3. If unsuccessful, go to 7
7. Return ‘no code found’. Go to 3.

3.3. Matching Human Coders

Two University of Pittsburgh physics domain knowledge experts each coded the same sample of error transactions, which were missing KC codes, and spanning four problems. Of these, only those transactions for which the experts had matching codes were used to evaluate the human-match standard. To compare how well codes from each heuristic matched those of the human coders, Kappa analysis was used.

Kappa provides a measure of the degree to which two judges, A and B, concur in their respective sorting of N items into k mutually exclusive categories. A 'judge' in this context can be an individual, a set of individuals who sort the N items collectively, or some non-human agency, such as a computer program or diagnostic test, that performs a sorting on the basis of specified criteria. The level of agreement is determined by the Kappa score. The closer the score is to 1, the better the agreement between pairs of codes [9].

Table 1: Results of the learning-curve standard

Criterion	LH_1	LH_2	TH_1	TH_2
AIC	6,444	6,285	6,759	6,653
BIC	7,578	7,414	7,894	7,781
Loglikelihood	-3,051	-2,972	-3,209	-3,155

Table 2: Results of the human-match standard

	LH_1	LH_2	TH_1	TH_2
Kappa Score	0.78	0.71	0.76	0.68
Asymp. Std. Error(a)	0.019	0.021	0.02	0.022
Approx. T(b)	66.7	61.7	63.5	58.2
Approx. Sig.	0.0	0.0	0.0	0.0
N of Valid Cases	527	527	520	521

a. Not assuming the null hypothesis

b. Using the asymptotic standard error assuming the null hypothesis.

4. Results

Table 1 shows the results obtained in fitting the data from each heuristic to equation (2). As can be seen, LH_2 was the leading performer in terms of data fit, with AIC

score of 6,285, BIC score of 7,414 and loglikelihood value of -2,972, where lower scores mean a better fit and correspondingly, a better cognitive model. Scores with a difference of greater than 6 are considered to be reliably different.

The better fit of LH_2 over LH_1 suggests that simply using the location-temporal method for error attribution may better characterize student learning patterns than using the intelligent tutor's attributions when present. For the temporal heuristic methods, we also see the simpler approach (TH_2) yielding a better fit than the tutor-informed approach (TH_1).

Table 2 shows the results of the human-match standard. To calculate Kappa scores, first, only observations where the two human raters agreed were selected. Then, the selected observations were matched against observations for each heuristic to compute Kappa scores.

As can be seen, the results are quite similar to the results of the learning-curve standard. In this case however, LH_1 and TH_1 provide a better fit than LH_2 and TH_2 respectively.

5. Discussion

Generally, the location heuristics has been shown to produce data with better fit to the learning curve standard and to human codes.

An interesting observation we made though, is that the heuristic models that coded all error transactions, that is, LH_2 and TH_2, produced better fitting data than LH_1 and TH_1 respectively, according to the learning curve standard. These results are shown in table 1.

Given that the human coders only provided a set of KC codes for each error transaction for which the Andes tutor failed to provide one, the results in table 2 are expected. While LH_1 and TH_1 coded the same transactions as the human coders, LH_2 and TH_2 coded more transactions since these methods coded all error transactions whether missing or not. As such, a greater mismatch was expected between the later heuristics and the human codes.

6. Conclusions

In this paper, we present and implement four automated heuristics for error attribution in order to facilitate learning curves analysis. We found that the location-temporal heuristics were better at predicting students' changes in error rate over time. The location-temporal heuristics also fit human codes better than the temporal heuristics. We intend to implement these heuristics on other datasets to test the generality of these results. The availability of datasets from the Pittsburgh Science of Learning Center's 'DataShop' (see http://learnlab.org) will facilitate the process of getting appropriate data.

Interestingly, we found in 2 of 4 comparisons (LH_2 > LH_1 and TH_2 > TH_1 for learning-curve standard) that two of the heuristic models proposed were better at error attribution than the original cognitive model of the intelligent tutoring system.

Overall, these results suggest that the heuristics proposed and implemented in this paper can generally aid learning curve analysis and perhaps, more generally, the design of student models.

Acknowledgements

This research was funded by the National Science Foundation grant SBE-0354420 to the Pittsburgh Science of Learning Center. We also wish to thank Hao Cen for his help with using the LFA tool.

References

[1] Anderson, , J. R., Bellezza & Boyle, C. F., (1993). The geometry tutor and skill acquisition. In J. R. Anderson (Ed.) Rules of the Mind, Chapter 8. Hillsdale, NJ: Erlbaum.

[2] Martin, B., Mitrovic, T., Mathan, S., & Koedinger, K.R. (2005). On Using Learning Curves to Evaluate ITS. Automatic and Semi-Automatic Skill Coding With a View Towards Supporting On-Line Assessment. Proceedings of the 12th International Conference on Artificial Intelligence in Education (AIED2005). Amsterdam, IOS Press.

[3] Koedinger, K.R. and Mathan, S. Distinguishing qualitatively different kinds of learning using log files and learning curves. In ITS 2004 Log Analysis Workshop. (2004). Maceio, Brazil. pp. 39-46.

[4] Cen, H., Koedinger, K. & Junker, B. (2005). Automating Cognitive Model Improvement by A* Search and Logistic Regression. In Proceedings of AAAI 2005 Educational Data Mining Workshop.

[5[Newell, A. & Rosenbloom, P. (1981). Mechanisms of Skill Acquisition and the Law of Practice. In Anderson J., (ed.): Cognitive Skills and their Acquisition, Erlbaum Hillsdale NJ

[6] Pirolli P & Wilson M. (1998). A Theory of Measurement of Knowledge Content, Access and Learning, Psychological Review vol.105, 1, pp. 58-82.

[7] Draney, K., Pirolli, P. & Wilson, M. (1995). A Measurement Model for a Complex Cognitive Skill. In Cognitively Diagnostic Assessment. Erlbaum, Hillsdale, NJ

[8] VanLehn, K., Lynch, C., Schultz, K., Shapiro, J. A., Shelby, R. H., Taylor, L., et al. (2005). The Andes physics tutoring system: Lessons learned. International Journal of Artificial Intelligence and Education, 15(3), 147-204.]

[9] Utah State University (2003). Kappa Tool User's Guide, http://www.gis.usu.edu/~chrisg/avext/downloads/kappa.pdf

[10] Wasserman, L. (2004) All of Statistics: A Concise Course in Statistical Inference. Springer

[11] Igbal A., Oppermann R., Patel A., & Kinshuk. (1999). A Classification of Evaluation Methods for Intelligent Tutoring Systems. In U. Arend, E. Eberleh & K. Pitschke (eds.). Software Ergonomie - Design von Informationswelten, B. G. Teubner Stuttgart, Leipzig, pp. 169-181.

[12] Corbett A.T., Anderson, J.R., O'Brien A.T., (1995). Student Modelling in the ACT Programming Tutor. In Cognitively Diagnostic Assessment. Erlbaum, Hillsdale, NJ

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Learning by Problem-Posing as Sentence-Integration and Experimental Use

Tsukasa Hirashima[a], Takuro Yokoyama[b], Masahiko Okamoto[c], Akira Takeuchi[b]
[a]*Hiroshima University, Japan,* [b]*Kyushu Institute of Technology, Japan,* [c]*Osaka Prefecture University, Japan*

Abstract. In this paper, "problem-posing as sentence-integration" is proposed as a new design of learning using computer-based method. We also introduce an interactive learning environment for the problem-posing and report experimental use of the environment in elementary schools. We have already developed a computer-based learning environment for problem-posing as concept combination. However, we could understand that it was difficult for students of the lower grades. Problem-posing by sentence-integration is a framework to realize learning by problem-posing in the lower grades. Sentence-integration is a simple method than problem-posing by concept combination, but it is expected to keep the learning effect to refine problem schema. The learning environment was used by 132 second grade students of two elementary schools to examine whether the lower grade students can pose problems in the environment or not. The effect of refinement of problem schema by the problem-posing was also analyzed by pre-test and post-test with excessive information problems. The students and teachers who were observers of the experiment accepted the learning environment as a useful learning tool to realize learning by problem-posing through the questionnaires and additional comments. Although the results suggest the effect of refinement of problem schema by the problem-posing, it was only compared against a no instruction control condition. Therefore, as a future work, it is necessary to compare the learning against other instructions to examine its characteristics.
tsukasa@isl.hiroshima-u.ac.jp

Keywords. Problem-Posing, Sentence-Integration, Interactive Learning Environment

Introduction

Learning by problem-posing is well recognized as an important way to learn mathematics [1-4]. However, despite the importance of problem posing, it is not popular as a learning method in reality. This is due to various factors. First, learners can make various kinds of problems, but some may be wrong and also some of the learners might repeatedly make similar problems, or make problems that are too simple to be useful for learning. In such cases, adequate feedback for each problem is required. However, because learners can make a large range of problems, it is difficult to prepare adequate feedback for every problem that learners might make. In problem posing, assessment of each posed problem and assistance based on the assessment is necessary. Because the above task puts a heavy burden on teachers, it is very difficult for them to use problem posing as a learning method. From this point of view, if a diagnosis function of problems posed by learners is realized, learning by problem posing will come to be used more.

Based on this consideration, we are investigating computer-based learning environment for interactive problem-posing that is composed of (1) problem-posing interface, (2) problem diagnosis function, and (3) help function for correcting or

completing the posed problems. We have already developed several learning environments for interactive problem-posing in arithmetical word problems [5, 6] and confirmed that exercise of problem-posing with the learning environment is effective to improve both problem-solving and problem-categorization abilities [7]. These results suggest that computer-supported interactive problem-posing is a promising approach to realize the learning by problem-posing.

One of the most important issues of these researches, however, is that the students who could pose problems with this learning environment are the fourth grade or above in elementary schools. Since learning of arithmetical word problems starts from lower grade, to realize learning by problem-posing in the lower grade is one of the most important issues. To realize such learning environment, then, it is necessary (1) to simplify the task of problem-posing, and (2) to keep the worth of problem-posing as a learning method.

Problem-posing in the previous environments is carried out as a combination of concepts. In the problem-posing, therefore, the process of making simple sentences by combining concepts and the process of making a problem by combining simple sentences are intermingled. The former process is corresponded to "transformation process" and the latter to "integration process" in a well-known process model of problem-solving of arithmetical word problems [9]. A simple sentence is interpreted linguistically in the transformation process, and the linguistic interpretations are integrated mathematically in the integration process. From the viewpoint of learning of mathematics, the integration process is far more important than the transformation process. Based on this consideration, we have proposed "problem-posing as sentence integration."

In the problem-posing as sentence integration, several simple sentences are provided to a learner. The learner, then, selects necessary sentences and arranges them in an appropriate order. Although the process to integrate simple sentences remains as the characteristics of problem-posing, the process to make simple sentences is simplified to the process to understand them. Therefore, the "problem-posing as sentence integration" is a promising approach to satisfy both (1) simplification of problem-posing task for lower grade students in the elementary schools, and (2) keeping the worth of problem-posing as a learning method. In other words, this is a problem-posing focused on sentence-integration process.

In this paper, in Section 1, learning by problem-posing is discussed in detail. In Section 2, a learning environment for problem-posing as sentence-integration is described. An experimental use of the learning environment in two elementary schools is also reported. Although the results suggest the effect of refinement of problem scheme by the problem-posing, it was only compared against a no instruction control condition. Therefore, as a future work, it is necessary to compare the learning against other instructions to examine its natures.

1. Learning by Problem-Posing

In our research of "Learning by Problem-Posing", the "problem" is defined as follows.

Problem = "Given-Information" + "Required-Information"

"Problem-Solving", is an activity to derive *required-information* from the *given-information* and then the *problem solving* should be able to complete deductively. "*Solution Method*" is, then, the method that is used in the *problem solving*. In the case of

an arithmetical word problem composed of the following three sentences {(I) *Tom had five pencils*. (II) *Ken received three pencils from Tom*. (III) *How many pencils does Tom have?*}, Sentence-I and –II are the given-information and Sentence-III is the required information. The solution method, then, is “5-3”. These definitions cover most of the problems used in problem exercises in arithmetic, mathematics, physics and so on. Based on this definition, problem-posing can be completed by adequately combining the following three elements: (1) given-information, (2) required-information, and (3) solution method. These elements, then, are used as constraints that a student has to satisfy in problem-posing. Problem-posing that dealt with in this paper is “solution-based problem-posing”. In solution-based problem-posing, a learner is required to make a problem and the problem should be composed of given-information and required-information that can be solved by a specified solution method. To realize this exercise, it is important to provide the way to let a learner to make the composition of given-information and required-information.

In the learning of arithmetical word problem, it is assumed that a learner has already had the ability to understand the meaning of each sentence in a problem and it is expected that the learner acquires the ability to understand the problem mathematically. In learning from problem-posing, therefore, it isn't indispensable to pose a problem by using natural language written by a learner. Based on these considerations, we have proposed three types of problem-posing methods as follows: (A) sentence template method [5], (B) problem template method [6, 7], and (C) sentence card method [8]. In this subsection, because of page limitation, the outline of only the sentence card method has been explained.

In the sentence card method, a learner is provided with several cards with a sentence in each card. The learner, then, is required to select some of them and to sort it out in a proper order. Although the process to make a sentence by combining concepts in other methods is substituted for the process to understand the sentences on the cards, the process to integrate simple sentences is same with the other problem posing methods. Therefore, the template card method is an approach to realize (1) to keep the worth of problem-posing as a learning method and (2) to simplify the task of problem-posing.

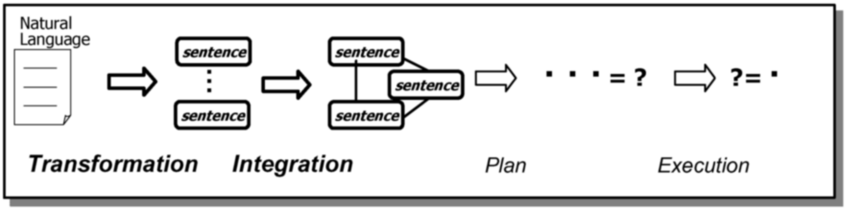


Figure 1. A Process Model of Problem-Solving of Arithmetical Word Problems.

A typical process model of problem-solving of word problems [9] is shown in Figure 1. The processes of *transformation* and *integration* are characteristics process of word problem. In the *transformation* process, natural language is interpreted linguistically. A linguistic interpretation of a natural language sentence is corresponds to “*sentence*” in Figure 1. In the *integration* process, they are integrated mathematically. The structure made by the *integration* process is often called “problem structure” and it is describes the understanding of the problem. Problem schema is mainly used in this process. From the viewpoint of learning of mathematics, the *integration* process is far more important than

the *transformation* process in the model. In the problem-posing by the sentence template method and the problem template method, *transformation* and *integration* are intermingled. Because the problem-posing with sentence card method is the problem-posing focused on the *integration* process, we call this type as "problem-posing as sentence integration". In the next section, a learning environment for interactive problem-posing as sentence integration, named MONSAKUN, is described.

2. Learning Environment for Problem-Posing: MONSAKUN

2.1 Interface for Problem-Posing

The interface of problem-posing in MONSAKUN is shown in Figure 2. The area in left side, imaged blackboard, is "problem-composition area". At the top of the area, a calculation expression is presented. A learner is required to pose a problem that is able to solve by the calculation expression. Here, the expression is the solution-method. Although the expression itself is easy, the learner has to consider the combination of not only a number but also a subject, object and predicate in each sentence. The three blanks in the area are the ones to put sentence cards. Sentence cards are presented at right side of the interface. A learner can move the card by drag&drop method freely in the interface. When a learner pushes "diagnosis button" under the problem-composition area, the system diagnoses the combination of sentences. The results of the diagnosis and message to help the learner's problem-posing is presented by another window.

In the case of Figure 2, the calculation expression is "5+3". To pose a problem that can be solved by this calculation, a learner has to put the cards of "Tom has five pencils", "Ken gave three pencils to Tom", and "How many pencils does Tom have?" into the blanks in this order. In this case, a learner sometimes selects "Ken received three pencils from Tom" instead of the second sentence. This is a typical error, and it is assumed that the cause is the association from "addition" as an operation to "receive" as a predicate.

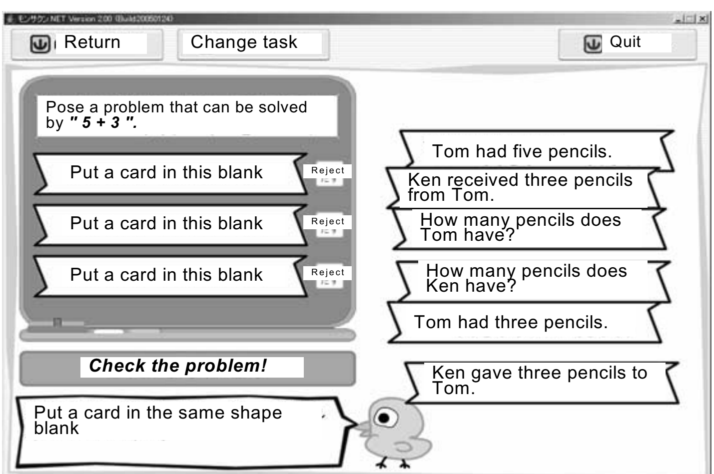


Figure 2. Problem-Posing Interface of MONSAKUN.

2.2 Diagnosis of a Posed Problem

A problem posed by a learner is diagnosed by the following three viewpoints: (1) problem type, (2) relations of concepts, and (3) numerical relation. The problem type is diagnosed by the combination of sentence type. For example, in the change problem, the first sentence has to describe a number of something exists, and the second sentence has to describe a change of the number. By this diagnosis, the type of problem the learner tried to pose can be detected. If the type can not be detected, that is, the combination of sentences is wrong, the system gives advises that what kind of sentence should be put in each blank.

When the problem type is detected, the relations of the concepts in the sentences can be diagnosed. Depending on each problem type, the conditions that the relations have to satisfy are different. For example, in the combination problem, the numbers should belong to the concepts that are able to add mutually, like the number of apples and the number of oranges. In the change problems, the object of the change in the second sentence should be appeared in the first and the third sentence. By using such conditions, the relations of concepts are diagnosed. When the problem satisfies the conditions, a calculation expression can be made from the problem. If the problem doesn't satisfy a condition of the problem type, the system judge the problem is wrong. Then, the system gives advises that what kind of condition should be specified.

In the diagnosis of numerical relation, a calculation expression is derived from the problem and then compared with the one that is the target of the problem posing. When the two expressions are the same, the posed problem is correct. If the derived expression is different only in the number, the system indicates the difference. When the expression is different in the operation, the system indicates that the posed problem is solved by the difference operation. When the calculation expression derives a minus number, the system indicates that the calculation is impossible because minus number has not been taught in elementary school. Since this case is happened only in subtraction, the system indicates that "it is impossible to subtract this number from that number".

3. Experimental Use of MONSAKUN

After trail uses of the learning environment by teachers in elementary schools and getting prospects that it might be usable and useful in learning of arithmetic for the second grade students, the environment was practically used in arithmetic classes. The purposes of the practice were to examine that (1) the students can pose problems by using the system, and (2) effect of the learning by problem-posing. Whether the students could pose problems in the environment was analyzed by using the logs of problem posing and the results of questionnaires. The learning effect was analyzed by the change of an experimental group and no instruction control group between pre-test and post-test for the student's scores of problem solving test of "extraneous problems" including an extraneous information that is not necessary to solve the word problems. A student who has sophisticated problem schema can perform well to solve them. This test, therefore, is useful to examine the learning effect for sophisticating problem schema by problem-posing as sentence integration. Comparison against other instruction condition is one of the most important future works.

3.1 Situation of the Experimental Use

The participants of our experiment are 132 students in the second grade of two elementary schools. They are divided into two groups: an experimental group and a control group. Because there are six classes originally, we assigned three classes to each group. The experimental group composed of 66 students was given a pre-test and a post-test before and after of the environment use. The control group composed of 66 students was given the pre-test and the post-test without the environment use.

In the experimental group, the pre-test was given at the beginning of the first class time of the first day. There was no task related to this use in the second day. In the third day, a series of two class times (One class = 45 minutes) were used to the use. Then, the post-test was given to them at the beginning of the first class of the fourth day. In control group, the pre-test was given at the beginning of the first class of the first day. There was no class related to problem-posing in the second and third day. Then, the post-test was given to them at the beginning of the first class of the fourth day. During this experiment, there was no holiday. Therefore, the difference between the experimental group and the control group is that whether they used the learning environment or not (that is, learning by the environment vs. no instruction). The control group also used the system for two class times after the post-test, in order to even up the learning opportunities following to the request of schools.

The experimental use was carried out in a computer room at each school, where every student was provided with one computer. In the experimental use, two staff supervised them and only answered them if there is any problem in the operations of the learning environment. To adjust the explanation time of the operations, time of answering the questionnaires, time of transfer from a normal classroom to the computer room and so on, the students used the systems for 50 minutes. The questionnaire was also given to the control group. In the analysis of the logs of system use and questionnaire, we don't distinguish the control group and the experiment group.

3.2 Results of System Use

In this subsection, we analyze whether the students can pose problems by using the learning environment based on the logs of the system use and the results of the questionnaire. Table 1 shows the results of the use. The number of "diagnosis requests" is the number that a student pushes "diagnosis button". This number can be regarded as the number of posed problems. When a problem that is required to diagnose is the same as the previously diagnosed problem, the request is not counted.

Table 1. The Results of the Use of MONSAKUN.

Total Time	Students	Posed Problems	Correct Problems
50 minutes	132	71 (in average)	50 (in average)

A student could pose 71 problems in average. In the posed problems, 50 problems were correct and 21 problems were incorrect in average. The student who uses the learning environment for the first time poses one correct problem in a minute. Therefore, the student can pose problems by using the environment. The number of incorrect problems suggests that the necessity of diagnosis function.

Table 2 shows the results of questionnaire. Most students enjoyed the problem-posing and interested to use MONSAKUN again. They also thought that the activity to pose problems was useful for their learning. The helps received from the system were also judged useful. Besides the teachers who were present at the class commented that most of the students kept their motivation to pose problems and their activities were valuable as arithmetic learning. These results suggest that MONSAKUN can be easily used by the second grade students and could be accepted as a useful learning tool by the both teachers and students.

Table 2. The Results of Questionnaires.

Answers / Questions	Yes	No	No idea
Did you enjoy the problem-posing?	131	0	1
Do you like to use the system again?	129	0	3
Do you think the problem-posing is important in learning?	120	5	7
Were the helps from the system useful?	121	4	7
Have you improved in posing problems?	123	1	8

3.3 Learning Effect

We used extraneous problems in both pre-test and post-test to access subject's problem solving performance. The extraneous problem includes extraneous information that is not necessary to solve the word problem. Figure 3 is an example of the extraneous problem. It is more difficult for children to solve the extraneous problem than to solve the standard word problem [10]. To solve the extraneous problem, children have to judge the relevance of sentences and find the sentence including the extraneous information. In this process, problem schema plays a crucial role. Therefore, the extraneous problem is useful to access children's problem schema.

> There are two apples.
> There are three oranges.
> <u>There are seven bananas</u>.
> How many apples and oranges are there in total?

Figure 3. An Extraneous Problem.

Both pre-test and post-test are composed of 12 extraneous word problems. We made two test versions and these two tests were used with counterbalance. For scoring the each problem, a correct numerical expression was given as 1 and an incorrect numerical expression was given 0 for each problems. The magnitude of score for both tests is 0 - 12.

Table 3 shows the mean scores of the pre-test and post-test as a function of condition and group. The difference of the mean scores between experiment group and control group was high (0.88), and the score of pre-test and post-test were correlated (0.83). A one way analysis of covariance, 2 (condition: experiment vs. control, covariance: pre-test) was carried out for the score of post-test. The trimmed means of the post-test in each condition, as the covariance is pre-test are 7.72 in the case of control and 8.55 in the case of experiment. The results revealed that the main effect of condition was significant ($F_{(1,129)}=84.71$, $p<0.5$) for the scores of post-test. This result indicated that the problem solving performance of the experimental group was higher than the control group in the post-test and the children in the experimental group have obvious gains from pre-test to

post-test in the learning with MONSAKUN. In conclusion, MONSAKUN improves the children's problem solving performance in the experimental group.

Table 3. Mean Scores of the Pre-test and Post-test.

Condition	Pre-test	Post-test
Experiment Group (n=66)	6.56 (SD=4.43)	8.21 (SD=4.01)
Control Group (n=66)	7.44 (SD=4.28)	8.06 (SD=4.03)

4. Conclusion Remarks

We have already developed several computer-based interactive environments for learning by problem-posing. However, it is often difficult to pose word problems in the environment for lower grade students. In this paper, we described a learning environment for learning by problem-posing as sentence-integration, in order to keep the worth of problem-posing as a learning method and also to simplify the task of problem-posing. In the problem-posing, several simple sentences are provided to the students. The students, then, select necessary sentences and arrange them in an appropriate order. Although the process to integrate simple sentences remains as the characteristics of problem-posing, the process to make simple sentences is simplified to understand them. Through experimental use of the learning environment in arithmetic classes, we confirmed that teachers and students accepted it as a useful tool for learning. Besides, analysis of scores of the pre-test and post-test as a function of condition and group suggests that the learning environment improve the children's problem solving performance in the experimental group. However, the result was derived by the comparison against a no instruction control condition. Therefore, as a future work, it is necessary to compare the learning against other instructions to examine its characteristics.

References

[1] Ellerton, N.F. (1986) Children's Made Up Mathematics Problems: A New Perspective on Talented Mathematicians. *Educational Studies in Mathematics*, Vol.17, pp.261-271.
[2] Brown, S.I., Walter, M.I. (1993) *Problem Posing: Reflections and Applica*tions, Lawrence Erlbaum Associates.
[3] Silver, E.A., CAI, J. (1996) An Analysis of Arithmetic Problem Posing by Middle School Students, *Journal for Research in Mathematics Education*, Vol.27, No.5, pp.521-539.
[4] English, L.D. (1998) Children's Problem Posing Within Formal and Informal Contexts, *Journal for Research in Mathematics Education*, Vol.29, No.1, pp.83-106.
[5] Nakano, A., Hirashima, T., Takeuchi, A. (1999) Problem-Making Practice to Master Solution-Methods in Intelligent Learning Environment, *Proc. of ICCE'99*, pp.891-898.
[6] Hirashima, T., Nakano, A., Takeuchi, A. (2000) A Diagnosis Function of Arithmetical Word Problems for Learning by Problem Posing, *Proc. of PRICAI2000*, pp. 745-755.
[7] Nakano, A., Hirashima, T., Takeuchi, A. (2002) An Evaluation of Intelligent Learning Environment for Problem Posing, *Proc. of ITS2002*, pp.861-872.
[8] Hirashima, T., Yokoyama, T., Okamoto, M., and Takeuchi A. (2006) A Computer-Based Environment for Learning by Problem-Posing as Sentence-Integration, Proc. of ICCE2006, pp.127-130.
[9] Kintsch, W., Greeno, J.G. (1985) Understanding and Solving Word Arithmetic Problem. *Psychological Review*, 92-1:109-129, 1985.
[10] Muth, K. D. (1992) Extraneous information and extra steps in arithmetic word problems. *Contemporary Educational Psychology*. Vol. 17, pp.278-285.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Dialog Convergence and Learning

Arthur WARD and Diane LITMAN
Learning Research and Development Center, University of Pittsburgh

Abstract. In this paper we examine whether the student-to-tutor convergence of lexical and speech features is a useful predictor of learning in a corpus of spoken tutorial dialogs. This possibility is raised by the Interactive Alignment Theory, which suggests a connection between convergence of speech features and the amount of semantic alignment between partners in a dialog. A number of studies have shown that users converge their speech productions toward dialog systems. If, as we hypothesize, semantic alignment between a student and a tutor (or tutoring system) is associated with learning, then this convergence may be correlated with learning gains. We present evidence that both lexical convergence and convergence of an acoustic/prosodic feature are useful features for predicting learning in our corpora. We also find that our measure of lexical convergence provides a stronger correlation with learning in a human/computer corpus than did a previous measure of lexical cohesion.

Keywords. Intelligent Tutoring, Learner Modeling, Discourse Analysis

1. Introduction

Human tutors have been shown to produce significantly larger learning gains than classroom instruction [1]. Because these human tutors teach using natural language dialog, many Intelligent Tutoring System (ITS) researchers are investigating the addition of natural language interfaces to their tutoring systems [2,3]. To help inform tutoring system design, many researchers search for aspects of natural language tutoring dialogs which might be associated with learning. Various types of dialog features have been investigated [4–6], many of which have demonstrated correlations to learning. Often, however, these features are difficult to implement in an ITS because they require hand coding of dialog transcripts. Shallow acoustic/prosodic [7], as well as other dialog features have been investigated which are automatically computable, but which may be less strongly related to learning depending on their linguistic sophistication [8]. As a result, there remains a need in the tutoring community for automatically computable dialog features which are also predictive of learning.

Work outside the tutoring community has suggested links between certain dialog characteristics and the amount of understanding that develops between dialog partners. In particular, Pickering and Garrod's Interactive Alignment Model [9] suggests a link between convergence and semantic alignment. In this paper we use the term "convergence" to mean when two dialog partners change some aspect of

their speech to be more similar to each other. We use the term "alignment" to mean when the internal representations they create become more similar to each other. "Priming," as described below, is one mechanism thought responsible for alignment and convergence.

The Interactive Alignment Model posits that each dialog partner processes speech in several levels. The speech signal is unpacked into low level phonetic and phonological representations, into lexical and syntactic representations, and so upward until high level semantic and situation model representations are determined. In this way, a number of internal representations of the incoming speech become active during processing. If speech is then produced while they are still active, these representations are more likely to be used than others that are less active. This priming process leads to alignment between the internal representations used at each level by the two dialog partners, and also to the convergence of their observable speech features. Priming is also thought to link neighboring levels of representation, so that alignment at one level increases the tendency to align at neighboring levels. We hypothesize that lexical and acoustic/prosodic (a/p) convergence between tutor and student is linked with alignment of their semantic representations, which may in turn be linked to learning.

Researchers have found that users will converge toward a dialog system on several lexical and a/p features. For example Coulston et al. [10], have found evidence for the convergence of spoken amplitude (loudness) toward that of a dialog agent. Bell et al. [11] have found that users will converge their speech rate toward that of a dialog system, and Brennan [12] has found that users will converge toward the lexical choices made by a system.

In section 2, we describe corpus measures of convergence which we developed in previous work [13]. In that work we showed that convergence was present in our corpus of tutoring dialogs with a human tutor. Here we show that our measures are also useful predictors of learning. In particular, we will show that our measure of lexical convergence predicts learning for the below mean pre-testers in both our human/human (hh) and our human/computer (hc) corpora. Lexical convergence is also a significant predictor of learning for all students in the hc corpus. One of our measures of a/p convergence will also be shown to predict learning among the high pre-testers in our hc corpora. Finally, we compare our measure of lexical convergence to our previous measure of lexical cohesion [14], and find convergence to have a stronger and more significant correlation with learning in our hc corpus.

2. Measuring Convergence

In [13] we built on previous work by Reitter et al. [15] to develop measures of lexical and a/p convergence which we applied to a corpus of tutorial dialogs. Both measures, lexical and a/p, proceed in the same general way. They both first locate a prime in tutor utterances, then define the next N student turns following this prime as a response window. For each distance d from the prime within the response window, they record the value of a response variable. After these data points have been collected for every prime in the corpus, linear regression is used to model the interaction between distance from the prime and the value

of the response variable. Following Reitter, a negatively sloped regression line is interpreted to mean an increased value of the response variable immediately following the prime, which then declines back toward it's global mean.

For the measure of lexical priming, the prime was defined to be the occurrence of some word w in a tutor utterance. Each word in the tutor utterance is treated as a potential prime, except, as described in [13], words which we classified as having had no alternative synonym. This adjustment was designed to make our measure better reflect priming's effect on lexical choice. In lexical priming, the response variable is the re-occurrence of the prime word in any of the following N student turns. For example, if the prime word was used again in the third student utterance following a prime, the data point [3,1] would be collected. In [13] we collected data sets for window sizes of 5, 10, 15 and 20 student turns, and fit lines to each of them using linear regression. The lexical slopes at the largest three window sizes were negative and significantly different than zero, which we argued was evidence for lexical priming effects.

We also looked for priming effects in six acoustic/prosodic (a/p) features: max, mean and min RMS (loudness) and max, mean and min f0 (pitch). For these a/p measures the prime was located wherever the tutor's value for the a/p feature in question was more than one standard deviation above the tutor's global mean. As with lexical priming, a response window was defined to be the next N student turns following the tutor's prime. The value of the a/p feature was recorded for each utterance in this window. For example, if the a/p feature in question was maxRMS (ie, maximum "loudness" in the turn), and the maxRMS value for the first student turn following the prime happened to be "1280," then the data point [1, 1280] would be collected. Linear regression produced slopes that were negative and significantly different from zero for max RMS at window sizes of 25 and 30, and for mean RMS at a window size of 30 student turns. Slopes were positive and significantly different from zero for min f0 at window sizes of 15, 20, 25 and 30 student turns. Significance thresholds were adjusted for multiple comparisons.

Both the lexical and a/p priming measures were validated by comparing their performance on naturally ordered vs. randomized data. The measures produced significant slopes on the naturally ordered data, but not on the randomized data. We argued that the lack of false positives on randomized data was evidence for the reliability of our measures. In [13], we fit regression lines to an entire corpus to show that priming effects could be detected in tutorial dialog. In the current work, we use the same method, but fit lines individually to each student in the corpus. We then show that the slope of the student's fitted regression line is a useful feature for predicting learning gains.

3. The Corpora

Our training set is the same corpus of tutoring sessions used in [13] to develop our measures of priming. In these tutoring sessions, a student first takes a pre-test to gauge knowledge of relevant physics concepts, then reads instructional material about physics. Following this, the human tutor presents a problem which the student answers in essay form. The tutor then examines this essay, identifies flaws

in it, and engages the student in a tutorial dialog to remediate those flaws. The student then reworks the essay, and the cycle repeats until the tutor is satisfied that all important points are covered. Each student finished up to ten problems. After the final problem, each student took a post-test, and learning gains were calculated. The resulting corpus contains sessions with fourteen students, totaling 128 dialogs, 6,721 student turns and 5,710 tutor turns. A mean pre-test split produces eight low and six high pre-testers.

Our testing corpus is a similar collection of tutoring dialogs, but collected using the ITSPOKE [16] spoken dialog tutoring system. ITSPOKE is a speech-enabled version of the WHY2-Atlas intelligent tutoring system [17]. The ITSPOKE system also teaches qualitative physics, and engages the student in the same cycle of dialog and essay revision as did the human tutor. In this corpus twenty students engaged in up to five tutorial dialogs each, resulting in a corpus of 95 dialogs, 2,335 student turns and 2,950 tutor turns. A pre-test split produces thirteen low and seven high pre-testers.

Our two corpora were similar in many respects, such as having similar subject matter and a similarly structured tutoring cycle. However, they were also different in two ways that may be relevant to the current study. First, the average number of student words per turn was higher with the human tutor. Students in the hh corpus averaged 5.21 words per turn, students in the human/computer (hc) corpus averaged 2.42 words per turn. Second, student utterances in the human tutor corpus seem to contain a broader range of words and have more complex syntactic structure. Student utterances in the computer tutor corpus, on the other hand, seem to be much more terse, containing more "keyword only" answers.

4. Finding Which Features Predict Learning

We want to find which, if any, of the measures of priming described in section 2 are associated with learning. To do this, we allow an automatic feature selection algorithm to select predictors on the hh corpus, then we test the validity of the selection by fitting a new linear model, which contains the selected features, to the hc corpus. If the features are significant predictors also in the second corpus, we take that to be evidence that the algorithm has selected useful features.

The automatic feature selection algorithm we use is "stepwise regression." It starts with an empty linear model, then adds and removes one variable at a time while attempting to minimize the Akaike Information Criteria (AIC) [18].

We first select features using all students, then separately for students with above-mean and below-mean pre-test scores. We do this because in previous work [14] our high and low pretesters responded differently to lexical features of the dialog. In addition, we will select features starting from several different initial feature sets. First we will allow the feature selection algorithm to choose among all ten features described in section 2. Then, we run feature selection again within each category. That is, we start with all features, then with only lexical features, then a/p features. We do this for two reasons. First, it will help avoid bias in feature selection, given that stepwise regression can produce different sets of predictors given slightly different starting points. Second, we are interested in

how the a/p features perform in isolation, because the lexical features depend on reliable speech transcriptions, which are often not available at runtime.

This approach resulted in nine runs of stepwise regression (3 student groupings x 3 initial feature sets). These nine runs produced three significant linear models. The remaining runs produced either empty models or models containing only "pretest" as a predictor. We test the usefulness of the selected predictors by using them to fit new models on our hc (human-computer) corpus.

						Pretest Only		Full Model	
		Student	Selected Features			Model	Adj	Model	Adj
	N	Group	Inter	preTest	lex.w20	pVal	R^2	pVal	R^2
1h	14	All hh	**0.484*** **	**0.652***	3.422	**0.012**	0.368	**0.017**	0.434
2h	8	hh low pre	0.331	**1.267***	**7.901***	0.29	0.047	**0.044**	0.597
1c	20	All hc	**0.315****	**0.583****	**-11.099****	**0.044**	0.162	**0.003**	0.431
2c	13	hc low pre	**0.866****	-1.075	**-17.361****	0.796	-0.08	**0.004**	0.589

Table 1. Lexical features selected on hh data (models 1h & 2h), tested on hc data (1c & 2c) Significance codes: p < .05: *; p < .01: **; p < .001: ***

						Pretest Only		Full Model	
		Student	Selected Features			Model	Adj	Model	Adj
	N	Group	Inter	preTest	meanRMS.w20	pVal	R^2	pVal	R^2
3h	6	hh hi pre	0.255	**0.998***	-0.015	**0.024**	0.694	**0.024**	0.860
3c	7	hc hi pre	**0.315***	0.816	**0.026****	0.089	0.363	**0.034**	0.725

Table 2. a/p features selected on hh data (model 3h),tested on hc data (3c)

The first of these three significant models was selected when starting with all features and all students. This is shown as model 1h in Table 1 (model numbers are shown in column one). In table 1 individual coefficients with p-values below .05 are shown in bold, with the level of significance indicated by asterisks (see caption). The model's selected features and their values can be read under the "Selected Features" columns. Model 1h can be read as: $posttestscore = .48 + .65 * pretest + 3.42 * lex.w20$. Lex.w20 is the slope of the lexical response line, fitted to each student using a window size of twenty. This model predicts that post-test score will increase .65 points for each point increase in pre-test score. Also, post-test score is predicted to increase by 3.42 points for each point increase in the lexical slope. Pre-test score is a reasonable selection, because it is correlated with post-test score in our human-human corpus (R = .64, pVal = .012). Lex.w20 is not itself significant in this model, but was selected because it improved the fit of the model by more than the AIC penalty for additional factors. The model p-value and adjusted R^2 given in the "Pretest Only" columns are for a model containing only an intercept and pre-test. The same numbers given in the "Full Model" columns are for a model which also includes the third predictor, in this case "lex.w20." The additional value of the lex.w20 predictor can be seen by comparing these two sets of numbers. In every case, the model's adjusted R^2 is larger for the full model. In almost every case, the model's p-value is better, as well.

The second significant model was selected when starting with all features and the low pre-test students, and is shown as model 2h of Table 1. This model contains the same features as were selected for model 1h, however in this model lex.w20 becomes individually significant.

Next, we test the features selected on the hh corpus by fitting them to our hc corpus. Model 1c of Table 1 shows the result of fitting the features of model 1h to the set of all hc students. The lex.w20 feature becomes highly significant here. Model 2c of Table 1 shows how the features of model 2h fare when fitted to the low pre-test students in the hc corpus. Lex.w20 is individually significant in this model, as it was in the hh corpus.

As described above, we also started feature selection from each category of features separately. Only one significant model was found in these runs, which is shown as Model 3h in Table 2. When starting with a/p (acoustic/prosodic) features and the hh high pre-testers, stepwise regression selects pre-test score and meanRMS.w20. MeanRMS.w20 is not individually significant when initially selected on the hh data. Model 3c in Table 2 shows the results of fitting the features from Model 3h to the hc data. The model as a whole remains significant on the hc data, while meanRMS.w20 becomes individually significant.

We have seen that features selected using only the human-human corpus perform well when fitted to a very different corpus of human-computer dialogs. This suggests that they may be genuinely useful indicators of learning in tutoring dialog. It is interesting to note, however, that the values fitted to these features differ widely between the corpora. When it is significant, the lexical priming feature "lex.w20," is fitted with positive coefficients in the human-human data, but with negative coefficients in the human-computer data. The meanRMS.w20 feature also is fitted with different signs in the two corpora, although in the hh corpus it is not individually significant[1].

Note that these features represent the slope of a line fitted to the student's response after a prime. A negative coefficient fitted to these features means that the student learns more as convergence increases and the slope becomes more negative. The negative lexical coefficients fitted to the hc corpus thus fit the predictions of the Interactive Alignment Model.

4.1. Lexical Cohesion vs. Convergence

As mentioned above, the usefulness of splitting our data by pre-test scores was suggested by previous work with a measure of lexical cohesion [14]. We now use that measure of lexical cohesion as a benchmark for comparing our current results. In [14] we measured cohesion by counting the number of cohesive ties between student and tutor, where a cohesive tie was defined as lexical overlap between consecutive speakers. Table 3 reproduces some of the results from that study. Rows two and three in Table 3 divide the students into above-mean and below-mean pre-test categories, exactly as in the present study. The center column shows the partial

[1]To test whether multicollinearity between window sizes might be preventing the selection of predictors which didn't switch sign between corpora, we also ran individual models for each predictor at each window size. All predictors kept their sign across different window sizes. We thank an anonymous reviewer for this suggestion.

correlation of cohesion and post-test, controlling for pre-test, for each group of students. The right column shows the significance of the correlation. Table 4 has the same layout, showing results for the partial correlation of the lex.w20 measure and post-test score, controlling for pre-test score. Both metrics are run on the same data set. Note that the significant lexical convergence correlations in Table 4 have consistently larger absolute magnitudes than the cohesion correlations in Table 3. Also, the convergence measure produces a significant correlation for the group of all students. This suggests that lexical convergence is a better predictor of learning than our previous measure of lexical cohesion.

HC Data	R	P-Value
All Students	0.431	0.058
Low Pretest	0.676	**0.011**
High Pretest	0.606	0.149

Table 3. Lexical Cohesion

HC Data	R	P-Value
All Students	-0.599	**0.005**
Low Pretest	-0.809	**0.001**
High Pretest	-0.437	0.327

Table 4. Lexical Convergence

5. Discussion and Future Work

The differences in coefficient polarity shown in Tables 1 and 2 are intriguing, and not yet explained. In section 3 we described several differences between the corpora which might be relevant. First the difference in student turn length between the hh and hc corpora may mean that our windows, while having the same size in turns, may be different in the number of words or amount of time they contain. Reitter et al. [15], on whose work we modeled our measures of convergence, experimented with defining response windows as spans of time. Possibly defining our windows similarly would remove the polarity differences between the corpora.

The other difference mentioned in section 3 was that there seems to be a difference in content, with the hh corpus containing more non-domain specific words. The difference in lexical polarity may then be partly explainable by differences in the words being used. Perhaps convergence is only positively correlated with learning when measured with domain-relevant tokens.

We have shown that measures of lexical and a/p convergence are useful features for predicting learning in two corpora of tutoring dialogs. When both are applied to the same corpus, our measure of lexical convergence also provides better p-values and larger correlations than our previous measure of lexical cohesion. Our other measure of convergence, which uses the "mean RMS" feature, was useful in predicting learning among our high pre-testers, and possesses the additional benefit of not requiring (machine or manual) transcription.

The corpora used in this study are our only two for which a/p features are currently available. Work is underway to generate a/p features for several additional corpora. Further experiments may help resolve the discrepancy in polarity of results reported here, by determining if fitted coefficients remain negative on future hc corpora. If experiments on these additional corpora are successful, we hope to include real-time measures of convergence in the ITSPOKE student model. A low or declining level of convergence may help indicate when the student is not aligning semantically with the tutor, and perhaps not learning.

6. Acknowledgments

This research is supported by the NSF (0325054), and by an Andrew Mellon Predoctoral Fellowship. We gratefully thank Tessa Warren for advice on this project, and also Joel Tetreault and the ITSPOKE group for many helpful comments.

References

[1] B. Bloom. The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. *Educational Researcher*, 13:4–16, 1984.

[2] M. Evens and J. Michael. *One-on-one Tutoring By Humans and Machines*. Erlbaum, 2006.

[3] Heather Pon-Barry, Karl Schultz, Elizabeth Owen Bratt, Brady Clark, and Stanley Peters. Responding to student uncertainty in spoken tutorial dialogue systems. In *International Journal of Artificial Intelligence in Education (IJAIED) Special Issue "Best of ITS 2004"*, volume 16, pages 171–194, 2006.

[4] A. Graesser and N. Person. Question asking during tutoring. *American Educational Research Journal*, 31:104–137, 1994.

[5] C. Rosé, D. Bhembe, S. Siler, R. Srivastava, and K. VanLehn. The role of why questions in effective human tutoring. In *Proceedings of AI in Education*, 2003.

[6] Mark Core, Johanna Moore, and Claus Zinn. The role of initiative in tutorial dialogue. In *10th Conference of the European Chapter of the Association for Computational Linguistics*, Budapest, Hungary, April 2003.

[7] K. Forbes-Riley and D. Litman. Correlating student acoustic-prosodic profiles with student learning in spoken tutoring dialogues. In *Proceedings Interspeech-2005/Eurospeech*, 2005.

[8] Kate Forbes-Riley, Diane Litman, Mihai Rotaru Amruta Purandare, and Joel Tetreault. Comparing linguistic features for modeling learning in computer tutoring. In *Proceedings 13th International Conference on Artificial Intelligence Education (AIED)*, Los Angeles, Ca., 2007.

[9] Martin J Pickering and Simon Garrod. Toward a mechanistic psychology of dialogue. In *Behavioral and Brain Sciences*, volume 27, 2004.

[10] R. Coulston, S. Oviatt, and C. Darves. Amplitude convergence in children's conversational speech with animated personas. In *Proceedings of the 7th International Conference on Spoken Language Processing*, 2002.

[11] Linda Bell, Joakim Gustafson, and Mattias Heldner. Prosodic adaptation in human-computer interaction. In *Proceedings of the 15th International Congress of Phonetic Sciences (ICPhS 03)*, Barcelona, Spain, 2003.

[12] Susan E. Brennan. Lexical entrainment in spontaneous dialog. In *International Symposium on Spoken Dialog*, pages 41–44, 1996.

[13] Arthur Ward and Diane Litman. Measuring convergence and priming in tutorial dialog. In *Technical report TR-07-148*, University of Pittsburgh, 2007.

[14] A. Ward and D. Litman. Cohesion and learning in a tutorial spoken dialog system. In *Proceedings of the 19th International FLAIRS Conference*, pages 533–538, 2006.

[15] David Reitter, Johanna Moore, and Frank Keller. Priming of syntactic rules in task-oriented dialogue and spontaneous conversation. In *Proceedings of the 28th annual Conference of the Cognitive Science Society*, Vancouver, 2006.

[16] D. Litman and S. Silliman. ITSPOKE: An intelligent tutoring spoken dialogue system. In *Companion Proc. of the Human Language Technology Conf: 4th Meeting of the North American Chap. of the Assoc. for Computational Linguistics*, 2004.

[17] K. VanLehn, P. Jordan, C. Rose, D. Bhembe, M. Boettner, A. Gaydos, M. Makatchev, U. Pappuswamy, M. Ringenberg, A. Roque, S. Siler, and Srivastava R. The architecture of why2-atlas: A coach for qualitative physics essay writing. In *Proc. 6th Int. Conf. on Intelligent Tutoring Systems*, pages 158–167, 2002.

[18] Hirotugu Akaike. A new look at the statistical model selection. *IEEE Transactions on Automatic Control*, AC-19(6):716–723, 1974.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Comparing Linguistic Features for Modeling Learning in Computer Tutoring

Kate FORBES-RILEY, Diane LITMAN, Amruta PURANDARE, Mihai ROTARU and Joel TETREAULT
University of Pittsburgh, Pittsburgh, PA, 15260, USA

Abstract. We compare the relative utility of different automatically computable linguistic feature sets for modeling student learning in computer dialogue tutoring. We use the PARADISE framework (multiple linear regression) to build a learning model from each of 6 linguistic feature sets: 1) surface features, 2) semantic features, 3) pragmatic features, 4) discourse structure features, 5) local dialogue context features, and 6) all feature sets combined. We hypothesize that although more sophisticated linguistic features are harder to obtain, they will yield stronger learning models. We train and test our models on 3 different train/test dataset pairs derived from our 3 spoken dialogue tutoring system corpora. Our results show that more sophisticated linguistic features usually perform better than either a baseline model containing only pretest score or a model containing only surface features, and that semantic features generalize better than other linguistic feature sets.

Keywords. Tutoring Dialogue Systems, Linguistics, Multiple Linear Regression

1. Introduction

Computer tutoring *dialogue* systems exploit natural language interaction in an attempt to improve student learning. A variety of features of natural language dialogue appear useful for modeling learning during tutoring. For example, longer student turns and higher percents of student words and turns were shown to correlate with learning in tutoring dialogues [1,2,3], as were specific dialogue acts (e.g., tutor feedback and question types) [1,2,4,5,6], discourse structure features, and local dialogue context features [4,7].

However, such features differ both in terms of how sophisticated they are linguistically and how easy they are to obtain. For example, turn counts represent surface linguistic properties that are easily computed by tutoring systems. Dialogue acts represent deeper pragmatic properties, but usually require manual labeling during system design or in a system corpus, even if automatic labeling is developed from the manual labeling.

We hypothesize that more sophisticated linguistic features will yield stronger models of student learning, making the extra labor to obtain them "worth the effort" from a system design point of view. These models can be used off-line to improve system dialogue design, or, if the model features are automatically computable in real time, on-line in adaptive systems that respond during tutoring. [8,9] support this hypothesis; e.g. in [8], students learned more from a system with more sophisticated dialogue feedback.

In this paper we examine this hypothesis, by comparing the utility of 6 linguistic feature sets (Section 2.2) for modeling learning in our tutoring system (Section 2.1). The feature sets represent different levels of linguistic sophistication and effort that our sys-

tem can automatically compute: 1) surface features, 2) semantic features, 3) pragmatic features, 4) discourse structure features, 5) local dialogue context features, and 6) all features combined. Prior work has not examined the relative utility of all these feature sets for modeling learning. We use the PARADISE framework [10] (Section 3), which is an application of multiple linear regression to dialogue system evaluation, to model learning. We train and test our models on 3 different train/test dataset pairs derived from our 3 spoken dialogue tutoring system corpora. Our results (Section 3.1) show that most sophisticated linguistic feature sets usually out-perform the baseline model and the surface feature model. We also find that semantic features generalize better than other linguistic feature sets, including those that are more labor-intensive to obtain.

2. Tutoring System Corpora and Linguistic Feature Sets

2.1. ITSPOKE System and Corpora

ITSPOKE (**I**ntelligent **T**utoring **SPOKE**n dialogue system) [4] is a *speech-enabled* version of the *text-based* Why2-Atlas tutoring system [11]. In ITSPOKE, the student first types an essay into a web interface answering a qualitative physics problem. The backend parses the essay and extracts a set of dialogue topics concerning misconceptions or incomplete explanations. ITSPOKE then engages the student in a dialogue that covers these topics. After the dialogue, the student revises the essay, thereby ending the tutoring for the problem or causing another round of dialogue/essay revision.

......spoken dialogue excerpt at 6.1 minutes after start of problem...

ITSPOKE$_4$: Good. Does the packet have an acceleration? If yes please specify its direction. *[ShortAnswerQ, Advance, Depth1, GoalRep0]*

STUDENT$_4$: horizontally forward yes *[Incorrect]*

> **ITSPOKE$_5$**: Well... To find the packet's acceleration we will use Newton's second law. What is the definition of this law? *[LongAnswerQ, Push, Depth2, GoalRep0]*
>
> **STUDENT$_5$**: an object in motion tends to stay in motion until it's act by an outside force *[Incorrect]*
>
> **ITSPOKE$_6$**: Well... Newton's second law says that the net force acting on an object equals the mass times the acceleration of the object. We can define Newton's second law as F=m*a. Hence what can we determine about the acceleration of the packet given that we know the net force acting on it is not zero and points straight down? *[DeepAnswerQ, Advance, Depth2, GoalRep0]*
>
> **STUDENT$_6$**: I don't know *[Can't Answer]*

ITSPOKE$_7$: Well... Since the net force F and the acceleration a are vectors, for the equation F=m*a to hold they both need to have the same direction. <*...more...*> Does the packet have an acceleration? If yes please specify its direction. *[ShortAnswerQ, PopUp, Depth1, GoalRep1]*

Figure 1. ITSPOKE Annotated Dialogue Excerpt

We use 3 ITSPOKE corpora for this study, referred to as: **S03**, **P05**, and **S05**. The user population of **S03** is different from **P05** and **S05**, because it was collected for a different purpose in a different year using a different recruitment method; all 3 corpora use slightly different ITSPOKE versions. The same procedure was used to collect the corpora: students with no college physics: 1) read a document of background material, 2) took a pretest, 3) worked 5 problems with ITSPOKE (each problem yields 1 dialogue), 4) took a posttest. The **S03** corpus contains 20 students. The **P05** corpus contains 27 students. The **S05** corpus contains 27 students. Figure 1 shows a corpus excerpt.

2.2. Linguistic Feature Sets

We computed 207 linguistic features per student in each corpus. By "per student" we mean each feature was calculated over all 5 dialogues of the student. Although all the features can be automatically computed by our system, they represent different levels of linguistic sophistication and effort. In addition, note that we computed all student turn-based features from a human transcription of the student turns (rather than the speech recognition output), both to estimate an upper bound on these features' usefulness for modeling learning, and to enable comparison with text-based tutoring systems.

1. Surface Feature Set: We computed 21 surface linguistic features per student; some have been used to model learning in other systems [1,2,3]. 18 features represent surface measures of student and tutor dialogue contribution. For each of the 6 groups in Figure 2, we computed 3 features: *Total Words, Total Turns, Average Words per Turn*. In addition, we computed 1 feature representing a temporal measure of total contribution: *Total Elapsed Time*, and 2 features representing speech recognition quality: *Total Timeouts* and *Total Rejections*[1]. These features all require minimal effort to compute.

S: Student spoken turns (transcribed)	**ST:** Student and tutor spoken turns
T: Tutor spoken turns (transcribed)	**SE:** Student contribution (spoken and essay)
E: Student typed essay turns	**STE:** Student and tutor contribution (spoken and essay)

Figure 2. Groupings Representing Student and Tutor Contributions

2. Semantic Feature Set: We computed 38 semantic features per student. 24 features represent semantic measures of student and tutor dialogue contributions. For each of the 6 groups in Figure 2, we computed 4 features: *Total Concepts, Average Concepts per Turn, Concept-Word Ratio, Total Unique Concepts*. Our notion of "Concept" distinguishes physics content words from other words in the dialogues. Our current method of concept extraction required some additional effort to implement: we count all words (concept tokens) in the student turns that appear in an online physics dictionary. This "Total Concepts" value is used to compute the average and ratio. "Total Unique Concepts" is computed by counting the number of *different* concepts (types) over all the student turns. For example, STUDENT_5 in Figure 1 contains 2 Unique Concepts: "motion" and "force", and 3 Total Concepts: "motion", "motion" and "force".

Our remaining 14 semantic features represent the correctness of the meaning underlying various surface forms of student answers (e.g., "down", "towards earth"). ITSPOKE automatically labels the correctness of recognized answers, although here we use a version of correctness obtained by feeding the transcribed answers into ITSPOKE[2]. We use 4 Correctness labels: *Correct*, *Partially Correct*, *Incorrect*, *Can't Answer*. *Can't Answer* is used for variants of "I don't know" (STUDENT_6, Figure 1). For each of the 4 Correctness labels, we computed a *Total*, a *Percent*, and a *Ratio* to each other label.

3. Pragmatic Feature Set: We computed 14 pragmatic features per student. 8 features are derived from automatically labeled dialogue acts, which required significant effort to implement. In particular, in prior work we first manually labeled the tutor Ques-

[1] A Timeout occurs when ITSPOKE does not hear speech by a pre-specified time interval. A Rejection occurs when ITSPOKE's confidence score for its recognition output is too low to accept that output.

[2] These two versions of correctness produce an agreement of 91% (0.84 Kappa). We have used various versions of correctness in prior studies [7,12,13].

tion Acts in the **S03** corpus [4].[3] These dialogue acts codify the intent underlying tutor questions. Our dialogues have a Question-Answer format; every ITSPOKE turn asks a question. Our labels, *Short, Long, and Deep Answer Question*, distinguish the type of ITSPOKE question in terms of its content and the type of answer it presupposes. *Repeat Question* labels variants of "Can you repeat that?" after rejections. From our manual annotations, we created a hash table associating each ITSPOKE question with a Question Act label, and used it to automatically label the ITSPOKE questions in the other 2 corpora. For each of the 4 Question Act labels, we computed a *Total* and a *Percent*.

Our remaining 6 pragmatic features are derived from a Goal Repetition variable conceived in prior work to track how often ITSPOKE goals repeat in a dialogue [12]. Each ITSPOKE question is associated with a goal, which repeats if it is not satisfied earlier in the dialogue (see ITSPOKE_7 in Figure 1). For each student, we compute a *Total* and a *Percent* over ITSPOKE goals repeated 0, 1, or 2 times (GoalRep0-GoalRep2).

4. Discourse Structure Feature Set: We computed 35 discourse structure features per student, which derive from significant effort in our prior work showing that the discourse structure Depth and Transition for each ITSPOKE turn can be automatically labeled [7]. The Depth labels (Depth1-Depth4) represent the depth of the turn's topic in the structure. E.g., $\text{ITSPOKE}_{4,7}$ in Figure 1 have Depth 1; $\text{ITSPOKE}_{5,6}$ have Depth 2. Here we computed a *Total* and a *Percent* for each of the 4 Depth labels.

The 6 Transition labels represent the ITSPOKE turn's position in the discourse structure relative to the previous ITSPOKE turn. The first ITSPOKE turn after an essay is labeled *NewTopLevel*. Each ITSPOKE turn at the same depth as the prior ITSPOKE turn is labeled *Advance* ($\text{ITSPOKE}_{4,6}$ in Figure 1). The first ITSPOKE turn in a sub-topic dialogue (which occurs after an incorrect answer to a complex question) is labeled *Push* (ITSPOKE_5). After a sub-topic dialogue completes, ITSPOKE pops up and either asks the original question again, labeled *PopUp* (ITSPOKE_7), or moves on to the next question, labeled *PopUpAdvance*. After student turn timeouts and rejections, respectively, ITSPOKE's question is repeated or some variant of "Can you repeat that?" is used, labeled *SameGoal*. Here we computed a *Total*, a *Percent*, and a *Ratio* to each other label, for each of the 6 Transition labels, yielding our remaining 27 discourse structure features.

5. Local Dialogue Context Feature Set: We computed 99 bigram features per student, representing local dialogue context. These features were derived from prior work that examined bigrams of discourse structure and correctness [7]. The bigrams consist of the labels for two consecutive turns; in particular, bigrams of: 1) Correctness labels, 2) Transition labels, and 3) a Transition label followed by a Correctness label. For example, in Figure 1, the $\text{Student}_4 \sim \text{Student}_5$ turn pair counts as an Incorrect$\sim$Incorrect bigram. The $\text{ITSPOKE}_4 \sim \text{ITSPOKE}_5$ turn pair counts as an Advance$\sim$Push bigram. The $\text{ITSPOKE}_4 \sim \text{Student}_4$ turn pair counts as an Advance$\sim$Incorrect bigram.

Thus we have 16 possible Correctness bigrams, 36 Transition bigrams, and 24 Correctness-Transition bigrams. However, in this study we excluded rarely-occurring bigrams whose total was less than 2 on average across our corpora to reduce overfitting in our learning models. For the remaining Correctness and Transition bigrams, we computed a *Total* and a *Percent* (the denominator is total speaker turns). For the remaining Transition-Correctness bigrams, we computed a *Total*, a *Percent* (the denominator is total student turns) and a *Relative Percent* (the denominator is the Transition Total).

[3] Our Acts are based on similar labels in related work [5]. Two annotators labeled these Acts in 8 dialogues in a parallel human tutoring corpus, yielding agreement of 90% (0.75 Kappa).

3. Student Learning Models

The PARADISE framework [14] uses multiple linear regression (MLR) to model a dialogue system performance metric (M) from an input set of dialogue features. The resulting model predicts M as the sum of n model parameters, p_i, each weighted by a coefficient, w_i. The model can then be used in system design, by treating p_i as a benefit to be maximized if w_i is positive, or as a cost to be minimized if w_i is negative.

Many PARADISE applications model "user satisfaction" (e.g., [10,14,13]). We have also used PARADISE to model learning [7,13], and MLR is used to model learning in other tutoring research (e.g., [6]). We model learning gain by predicting posttest and forcing pretest as the first model parameter; the stepwise procedure automatically selects other features for inclusion in the model. However, as in [14], the procedure can only select features that correlate with posttest controlled for pretest (at $p \leq 0.1$); this can help prevent overfitting and reduce the average error of the model.

Here we train a learning model on each of our 5 linguistic feature sets, and from all feature sets combined, to examine their relative usefulness. As a baseline, we train a model with only pretest. Further, we train the 7 models on 3 different datasets corresponding to all combinations of 2 corpora: **S03+P05**, **S03+S05**, and **P03+S05**. In each case, we test the models on the third corpus. This allows us to investigate how well the models perform and generalize with different user populations and system versions.

3.1. Results

Table 1 shows our 7 learning models trained on the **S03+P05** corpora and tested on the **S05** corpus. The first column shows the feature set used to build the model. The second column shows how many features in each set correlated and were included in the stepwise procedure. The third column presents the trained model: each selected parameter and its weight; larger weights indicate parameters with greater relative predictive power.[4] The last two columns show how well the model performed on the training and test data, respectively, in terms of the amount of Posttest variance accounted for by the model (R^2). For example, the model trained on Surface features contains Pretest and Total Elapsed Time, and outperforms the Pretest model on the training data, but not on the test data.

Considering training, all models outperform the Pretest model. The Semantic, Local Context[5], and All models outperform the Surface model. The All model performs best, and only contains Semantic and Local Context features. The Pragmatic and Discourse Structure feature sets performed worst. No Pragmatic features correlated at $p \leq 0.1$; one feature correlated at p=.12 and was forced into the model for comparison.

Considering testing, only the Semantic model outperforms the Pretest model, and Pretest is the strongest predictor in all models. This suggests both that Pretest is our most generalizable predictor of Posttest, and that of our linguistic features, Semantic features are the most generalizable. However, all sophisticated linguistic models outperform the Surface model. This suggests that sophisticated linguistic features are "worth the effort" and that it would be worthwhile to expend further effort extending these feature sets to find more generalizable features. Finally, the Semantic model outperforms the All model, suggesting that other procedures may be better than stepwise for developing an "absolute" best learning model.

[4] These weights are the standardized coefficients (beta weights), i.e. based on scaling of the input features.

[5] A fourth parameter in this model was excluded, because it was highly correlated with another parameter ($R \geq .70$); inclusion of highly correlated parameters can affect the coefficients [14].

Table 1. Learning Models Trained on S03+P05 (47 Students) and Tested on S05 (27 Students)

Feature Set	# Feats	Learning Model	train R^2	test R^2
Pretest	n/a	0.57*Pretest	0.319	0.377
+Surface	4	0.70*Pretest + 0.31*Elapsed_Time#	0.396	0.293
+Semantic	20	0.48*Pretest + 0.33*S_Concept/Word + 0.22*E_Unique_Concept#	0.501	0.386
+Pragmatic	1 (p=.12)	0.62*Pretest + 0.20*LongAnswerQ#	0.356	0.323
+Disc. Str.	3	0.46*Pretest - 0.27*PopUp/Advance	0.378	0.323
+Local	15	0.58*Pretest - 0.34*PopUp∼Incorrect% + 0.34*Advance∼Advance#	0.531	0.330
+All	42	0.51*Pretest + 0.41*Advance∼Advance# - 0.33*PopUp∼Incorrect# - 0.31*PopUp∼Advance% + 0.29*STE_Concept/Turn	0.673	0.321

Table 2 shows our results for training on **S03+S05** and testing on **P05**. This training set and the prior one combine the 2003 and 2005 user populations; thus we expected them to yield similar results and more generalizable models.

Table 2. Learning Models Trained on S03+S05 (47 Students) and Tested on P05 (27 Students)

Feature Set	# Feats	Learning Model	train R^2	test R^2
Pretest	n/a	0.54*Pretest	0.288	0.452
+Surface	3	0.56*Pretest + 0.23*SE_Words#	0.341	0.356
+Semantic	18	0.51*Pretest + 0.32*S_Concept/Word	0.392	0.524
+Pragmatic	1 (p=.11)	0.48*Pretest - 0.21*RepeatQ#	0.330	0.500
+Disc. Str.	6	0.58*Pretest - 0.26*PopUp/Push	0.352	0.278
+Local	20	0.60*Pretest - 0.32*PopUp∼Incorrect% + 0.30*Push∼Push%	0.482	0.304
+All	47	0.64*Pretest - 0.33*PopUp∼Incorrect% + 0.26*STE_Concept/Word + 0.23*Push∼Push%	0.541	0.375

Considering training, we see considerable similarity with the prior table. Again all linguistic models outperform the Pretest model, and the Semantic, Local Context, and All models outperform the Surface model; however, here the Discourse Structure model does too. Again the All model is best and contains only Semantic and Local Context features. The Pragmatic set again had one feature forced into the model for comparison.

Considering testing, we continue to see considerable similarity with the prior table. The Semantic model again outperforms the Pretest model, and contains the two of the same parameters as its counterpart in the prior table. However, here the Pragmatic model also outperforms the Pretest model, which suggests that using only correlated features may not produce an “absolute” best model. Again no other models outperform the Pretest model. Again the Semantic, Pragmatic, and All models again outperform the Surface model, although here the Discourse Structure and Local Context models do not. How-

ever, the All model contains the features of the Local Context model plus one additional Semantic feature, and these Local Context features are derived from bigrams of Discourse Structure features. Overall our results in these tables suggest both that Semantic features are our most generalizable linguistic predictors of Posttest, and that our other more sophisticated linguistic feature sets are probably "worth the effort" too.

Table 3 shows our results for training on **P05+S05** and testing on **S03**. Since this training set includes only 2005 students, we expected less similarity and generalizability.

Table 3. Learning Models Trained on P05+S05 (54 Students) and Tested on S03 (20 Students)

Feature Set	# Feats	Learning Model	train R^2	test R^2
Pretest	n/a	0.64*Pretest	0.412	0.214
+Surface	1	0.59*Pretest - 0.21*Rejections#	0.452	0.259
+Semantic	8	-0.40*Incorrect/Correct + 0.35*Pretest + 0.24 S_Concept/Word	0.585	0.338
+Pragmatic	2	0.53*Pretest + 0.27*GoalRep0%	0.470	0.135
+Disc. Str.	7	0.55*Pretest - 0.23*Depth4%	0.455	0.145
+Local	28	0.40*Pretest - 0.31*PopUp∼Incorrect# + 0.28*Can'tAnswer∼Correct%rel	0.567	0.284
+All	46	-0.40*Incorrect/Correct + 0.35*Pretest + 0.24 S_Concept/Word	0.585	0.338

Considering training, we see more similarity than expected. Again all linguistic models outperform the Pretest model; they all also outperform the Surface model. Again the All model performs best; here however it is identical to the Semantic model and a Semantic feature is the strongest predictor. Furthermore, Student Concept-Word Ratio was consistently selected by the Semantic models in all tables, suggesting it is our most generalizable linguistic feature. The Local Context model is again second best.

Considering testing, performance as expected is generally lower as compared to the other tables, but we continue to see similarities. Although here all but 2 models (Pragmatic and Discourse Structure) outperform both the Pretest and Surface models, as in the prior tables, the Semantic model continues to performs best on testing.

4. Conclusions and Current Directions

This paper examined the relative usefulness of linguistic feature sets for modeling learning in computer dialogue tutoring. We hypothesized that more sophisticated linguistic feature sets would yield stronger learning models, making them "worth the effort" to obtain. We built models from 6 different linguistic feature sets on 3 different subsets of our 3 ITSPOKE corpora, then compared how the models generalized to new test sets.

Our results support our hypothesis. The Semantic models consistently outperformed all models on testing, yielding our most generalizable linguistic features, with Student Concept-Word Ratio present in every Semantic model and some Semantic feature(s) in every All model. The Semantic and Local Context models consistently outperformed the Surface model on testing. The performance of the Discourse Structure and Pragmatic models was more variable, but the Pragmatic model outperformed the Surface model on testing in two datasets, while in every dataset either the Discourse Structure model outperformed the Surface model on training, or these features appeared in models (Local or All) that did so. Moreover, no Surface features were selected in any of the All models.

Although our linguistic features can all be computed automatically, more sophisticated features required more effort to conceive and implement than surface features. Our results suggest it would be worthwhile to expend further effort to find more generalizable sophisticated features. For example, we are extending our Semantic feature set to include de-stemmed and synonymous concept words and phrases. We are extending our Local Context features to include other bigrams. We will also add features based on recognized student speech, as well as para-linguistic features, such as affective states.

We will also try other model-building procedures. We used PARADISE based on our prior work in dialogue systems; this can be viewed as one method of data mining for learning-related features. Although our models do not imply causality between model parameters and learning, they suggest hypotheses about how learning can be increased, i.e. by modifying the system to try to minimize negative parameters (costs) and maximize positive parameters (benefits). These hypotheses can be tested by evaluating the new system. For example, the Surface Feature model in Table 3 suggests rejections are a cost. We can test this by lowering ITSPOKE's recognition threshold so it rejects answers less frequently. Incorporating our most generalizable parameters into system design will likely have the most impact on learning in new user populations. However, the hypotheses suggested by these parameters do not always suggest obvious system changes. For example, we cannot directly manipulate the Student Concept-Word Ratio; increasing this ratio may involve changing the system's response during the tutoring.

Acknowledgements

NSF (#0325054 and #0328431) and ONR (N00014-04-1-0108) support this research.

References

[1] M. G. Core, J. D. Moore, and C. Zinn. The role of initiative in tutorial dialogue. In *Proc. EACL*, 2003.

[2] C. P. Rosé, D. Bhembe, S. Siler, R. Srivastava, and K. VanLehn. The role of why questions in effective human tutoring. In *Proceedings of AIED*, 2003.

[3] S. Katz, D. Allbritton, and J. Connelly. Going beyond the problem given: How human tutors use post-solution discussions to support transfer. *Internat. Jnl of Artificial Intelligence and Education*, 13, 2003.

[4] D. J. Litman and K. Forbes-Riley. Correlations between dialogue acts and learning in spoken tutoring dialogues. *Journal of Natural Language Engineering*, 12(2):161–176, 2006.

[5] G. Jackson, N. Person, and A. Graesser. Adaptive tutorial dialogue in AutoTutor. In *Proceedings Workshop on Dialog-based Intelligent Tutoring Systems at ITS*, 2004.

[6] M. T. H. Chi, S. A. Siler, H. Jeong, T. Yamauchi, and R. G. Hausmann. Learning from human tutoring. *Cognitive Science*, 25:471–533, 2001.

[7] M. Rotaru and D. Litman. Exploiting discourse structure for spoken dialogue performance analysis. In *Proceedings of EMNLP*, Sydney, Australia, 2006.

[8] B. Di Eugenio, D. Fossati, D. Yu, S. Haller, and M. Glass. Natural language generation for intelligent tutoring systems: a case study. In *Proceedings of AIED*, The Netherlands, 2005.

[9] M. Evens and J. Michael. *One-on-One Tutoring by Humans and Computers*. Lawrence Erlbaum,, 2006.

[10] M. Walker, D. Litman, C. Kamm, and A. Abella. PARADISE: A general framework for evaluating spoken dialogue agents. In *Proceedings ACL/EACL*, pages 271–280, 1997.

[11] K. VanLehn, P. W. Jordan, C. P. Rosé, D. Bhembe, M. Böttner, A. Gaydos, M. Makatchev, U. Pappuswamy, M. Ringenberg, A. Roque, S. Siler, R. Srivastava, and R. Wilson. The architecture of Why2-Atlas: A coach for qualitative physics essay writing. In *Proceedings of ITS*, 2002.

[12] J. Tetreault and D. Litman. Comparing the utility of state features in spoken dialogue using reinforcement learning. In *NAACL*, New York City, 2006.

[13] K. Forbes-Riley and D. Litman. Modelling user satisfaction and student learning in a spoken dialogue tutoring system with generic, tutoring, and user affect parameters. In *Proceedings HLT-NAACL*, 2006.

[14] M. Hajdinjak and F. Mihelic. The PARADISE evaluation framework: Issues and findings. *Computational Linguistics*, 32(2):263–272, June 2006.

This page intentionally left blank

Learner Modelling

This page intentionally left blank

Comparing Student-Constructed Open Learner Model Presentations to the Domain

Andrew MABBOTT and Susan BULL
Electronic, Electrical and Computer Engineering,
University of Birmingham, Edgbaston, Birmingham, B15 2TT, U.K.
{axm891, s.bull}@bham.ac.uk

Abstract. Increasingly, learning environments are opening the learner model to the user it represents. This paper describes a study in which students were able to create their own presentations of their learner model. We examine the nature of the learner model views created, and the degree of correspondence with the domain. Our results suggest learners have different preferences for the type of view they would like to create, and in many cases use informal or unstructured representational systems.

Introduction

Learner models have been externalised to the user using a variety of presentations (e.g. [1,2,3,4]), with one of the aims being to enhance reflection. Multiple representations have been argued to be effective in fostering deeper understanding of a task, as they can allow the use of complementary information processing methods and support individual differences [5]. While users of multiple-view open learner models have been found to have individual preferences for representation [6,7,8], the available views may not necessarily allow the learner to interact with the model in the most appropriate way *for them*, and a logical step may be to allow the learner to construct their own representation. Self-constructed representations potentially allow information to be usefully reorganised, aiding the monitoring of progress, highlighting problem areas, and facilitating self-explanations [9]. Some attempts have been made to allow the learner to influence the presentation of the learner model: STyLE-OLM [3] explicitly models understanding of conceptual relationships, with the task of constructing the model central to the interaction; in VisMod [4], the learner more simply selects the dimension (colour, size, length) for each of the model properties. This paper aims to investigate the type of representations learners construct in a more open scenario, and whether these representations are an accurate depiction of the domain. We present results from a study where students were able to create their own presentations of the learner model according to two styles of layout: tree and map.

1. The Flexi-OLM Open Learner Model

Flexi-OLM [7] is designed to prompt learners to reflect on their understanding of C Programming. Multiple-choice and short-answer responses contribute to a personal

learner model, which can be viewed in seven different presentations, corresponding to four types of layout: tree (topic hierarchy and lecture outline); map (concept map and prerequisite layout); list (alphabetical index and list ranked according to knowledge); and textual. The views convey the same information – proficiency on individual topics – via a consistent colour scheme: well understood topics are coloured green and poorly understood topics white, with two intermediate levels indicated by shades of yellow. Red signifies potential misconceptions while grey denotes a lack of information (i.e. insufficient questions answered). Views differing in abstract structure may have similar relational structure, with the hierarchy and concept map based on conceptual relationships and the lectures and prerequisites views indicating sequences (see Figure 1). While all views contain the 43 learner model topics, some are augmented with meta-topics where knowledge is not modelled, but which allow the structure to make sense (e.g. 'operators', 'lecture 4' etc.). In the concept map, labelled arrows illustrate propositional relationships, while the prerequisites view uses unlabelled arrows and lines to model prerequisite and co-requisite relationships respectively.

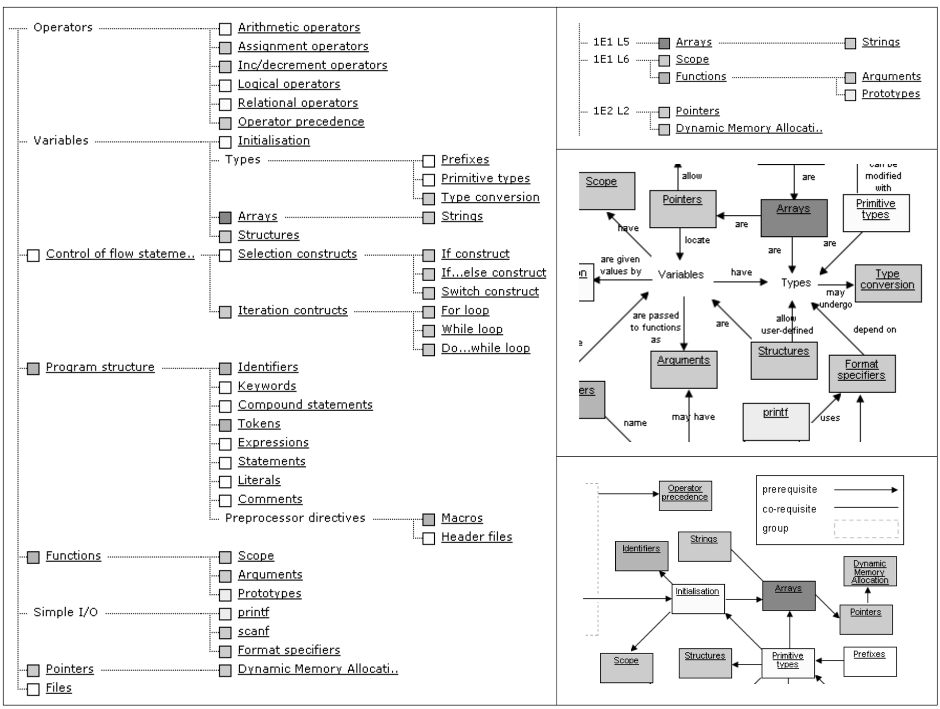


Figure 1: The hierarchy view plus excerpts of (top-bottom) the lectures, concept map and prerequisites views

For the study, 'tree' and 'map' views were added, in which users could create their own layout. (Since the list view is constrained to a simple sequencing of topics, allowing less potential for creative layout, we focus on use of the tree and map views.) The views contain a list of all topics in the learner model and a blank area onto which the topics may be dragged and arranged (see Figure 2). 'Custom' topics may be added, although knowledge level is not represented for these. In the tree view, any topic

dragged onto another becomes a child of that topic, while topics may be positioned anywhere in the map view and linked with labelled arrows if necessary.

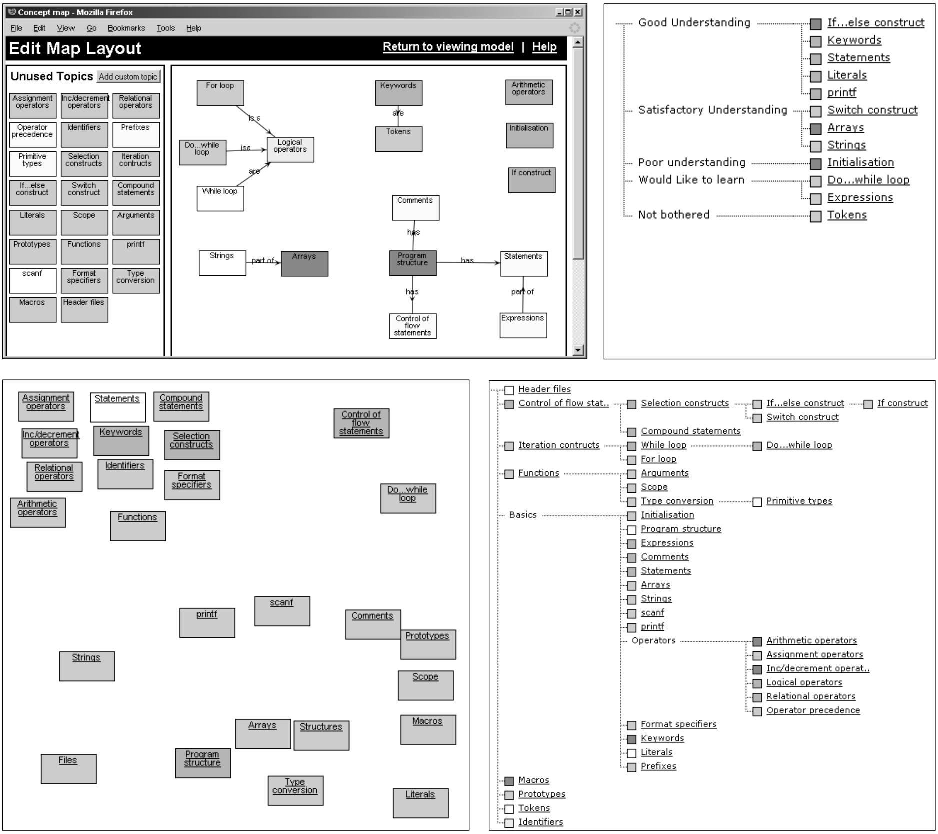


Figure 2: Representation construction interface with examples of learner-constructed views

2. What Kind of Learner Model Presentations do Users Create?

A study was carried out into the types of presentations users create when given the opportunity to construct their own views of the learner model. We investigate two issues: the kind of information learners are trying to convey, and whether the views created are accurate according to the domain.

2.1. Participants, Materials and Method

36 third-year undergraduate students participated in the study as part of a course on interactive learning environments, while a further 6 (3 MSc, 3 PhD) also volunteered, making a total of 42. The students had prior experience of a simple open learner model system [6]. After dividing into two groups, participants began by answering questions to allow Flexi-OLM to build a learner model. Group A were then invited to create views of their learner model data using the two templates (tree, and map), while

Group B used the predefined views. Later, each group switched to the opposite set of views. Students were not instructed on the type of views to create, but instead asked to arrange each of the self-constructed layouts in the way that suits them best.

To facilitate an analysis of the meaningfulness of user-created views, all 903 possible pairings of the 43 domain topics were analysed by a C Programming expert and classed as either strongly related, weakly related or unrelated. The exact nature of the relationship was not specified as this may be somewhat subjective.

2.2. Results

Figure 2 shows examples of the learner-constructed learner model views.

2.2.1. The Map View

Several students made little or no use of the map view, while the majority did not use the facility to link topics with labelled arrows, and only seven employed more than two such links in their layout (See Table 1). Students using the predefined views first (Group B) appear to create slightly more detailed maps than those beginning with the self-constructed views (Group A). Only four students used custom topics in their map.

Table 1: Number of participants vs. number of topics and links used in map view

	Number of topics			**Number of links**		
	Group A	**Group B**	**Total**	**Group A**	**Group B**	**Total**
None	3	3	6	14	9	23
1-2	3	0	3	5	7	12
3-5	7	7	14	2	2	4
6-10	3	4	7	0	2	2
More than 10	5	7	12	0	1	1

Table 2 examines the nature of the links created, with each column representing an individual user, and rows corresponding to categories of link. Conceptual relationships were classed as propositional (e.g. 'logical operators - *are* - operators'), non-propositional (e.g. 'if-else construct - *constructs* - if construct'), or incorrect (e.g. 'for loop - *is a* - logical operator'). Relationships classed as non-conceptual related to planning (e.g. '*spend more time'*), level of difficulty (e.g. '*easy*'), or understanding (e.g. '*learnt*'), while the remainder expressed no apparent relationship.

Table 2: Nature and quantity of links in map view

Link Type	**Group A**							**Group B**											
Non-conceptual relationship		2									1						1		39
Propositional conceptual relationship																		6	
Non-propositional conceptual relationship						3						1		2	2		6		
Incorrect conceptual relationship															1			3	
No relationship	1		2	2	2	1	4	1	1	1			2			4	1		
Total links	**1**	**2**	**2**	**2**	**2**	**4**	**4**	**1**	**1**	**1**	**1**	**1**	**2**	**2**	**3**	**4**	**8**	**9**	**39**

Only one student created a map using propositional conceptual relationships, while two others used mainly correct non-propositional relationships, and three used exclusively planning/knowledge related links. Around half made only links which indicated no relationship. More students from Group B created meaningful links.

Some students appear to have clustered together related topics without linking them. For each topic in a user's layout, the closest topic was determined, with preference given to those connected with an explicit link. The strength of the connection between each of these pairs of topics was recorded. Figure 3 indicates the number of students having a particular percentage of adjacent topics which are related, while Figure 4 considers only strongly related topics. The nine students using fewer than three topics were excluded from this analysis (at least three topics are required to indicate clustering since two topics would be adjacent by default; three allowed inclusion of students who were concentrating on a small area of their model).

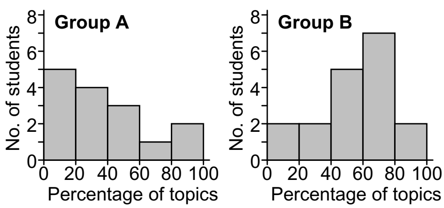


Figure 3: Topics related to nearest topic

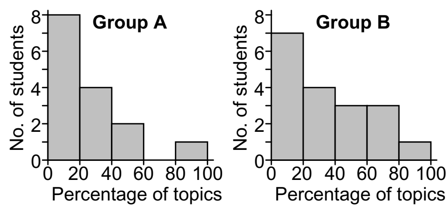


Figure 4: Topics strongly related to nearest topic

A large number of students created maps where the majority of adjacent topics were conceptually unrelated and few constructed layouts with a strong conceptual relationship between a majority of topics. Students using the predefined views before creating their own appear more likely to group conceptually related topics.

Figure 5 illustrates the percentage of related adjacent topics for stronger ability students (those above the median according to the learner model) compared to weaker ability students. Level of understanding of the domain does not appear to have had an impact on whether students positioned related topics adjacent to each other.

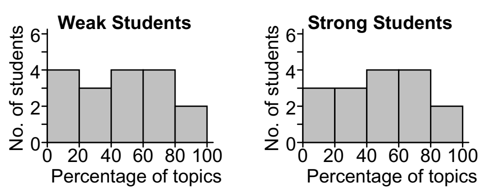


Figure 5: Topics related to nearest topic for strong and weak students (Group A and B combined)

2.2.2. The Tree View

Table 3 shows the number of topics used in the tree view. As with the map view, several students made no use of the tree view or added only one or two topics to it. While fewer students in Group B used the tree view, there were also more students who created trees using a larger number of topics. Seven students created custom topics, with the most created being five. Table 4 shows the number of levels used in the hierarchies created by students. A number of students created linear structures either with all topics at the root level, or with each topic having only one child. The highest number of levels was 5 (the predefined hierarchy and lectures views have 3 and 4 levels respectively).

Table 3: Number of topics in tree layout

No. of Topics	Group A	Group B	Total
None	2	5	7
1-2	3	2	5
3-5	8	3	11
6-10	4	8	12
More than 10	4	3	7

Table 4: Tree depth

Depth	Group A	Group B	Total
0	2	5	7
1	7	3	10
2	6	7	13
3	3	3	6
4	1	2	3
5	2	1	3

An analysis of the relationship between child and parent topics was carried out on the 25 layouts with more than one level. Custom parent topics were classified as either conceptual (e.g. 'operators') or related to planning or knowledge (e.g. 'to learn', 'don't understand'). Model topics were classed as strongly related, weakly related, or unrelated, according to the previous definitions. Table 5 indicates the categories accounting for the majority of the topics in each user's tree layout.

Table 5: Most common relationship to parent topic in tree layout

Majority category		Group A	Group B	Total
Custom (Planning / knowledge)		1	2	3
Custom (Conceptual)		0	2	2
Related	Mostly strongly	3	4	7
	Mostly weakly	0	2	2
Unrelated		8	3	11

Three users constructed layouts based on level of understanding or learning sequence while two used custom topics to create a conceptual hierarchy. Of the remainder, there were 9 cases where the majority of topics were conceptually related to the parent and 11 where the majority were unrelated. Students constructing their own views before using the predefined views were more likely to associate a topic with a conceptually unrelated parent. Since simply associating related topics does not guarantee that the relationship is hierarchical, each link was analysed and classed as either hierarchical or non-hierarchical. Figure 6 shows the proportion of topics having a hierarchical relationship to the parent, for each user in Group A compared to Group B, and Figure 7 presents a similar comparison between the strong and weak students. Around a third of the students constructed layouts where almost all of the relationships were hierarchical, while the remainder constructed maps where few were. This did not appear to be affected by whether students began with the self-constructed or predefined views, or by level of understanding according to their learner model.

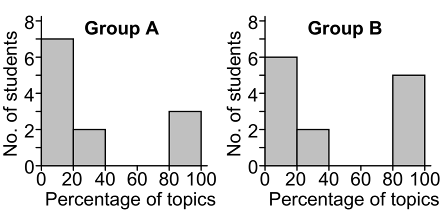


Figure 6: Percentage of topics with hierarchical relationship to parent (Group A / B)

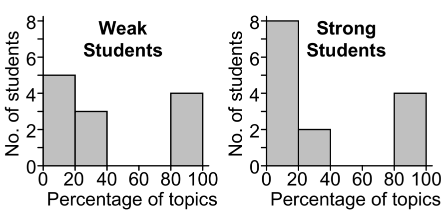


Figure 7: Percentage of topics with hierarchical relationship to parent (strong/weak students)

2.3. Discussion

An interesting variety of user-created layouts was observed, suggesting differences in the types of learner model view students would like to use. This may reflect preferences for representational system, or may be dictated in part by the task an individual is working on. For example, recognising problem areas may be easier with a particular kind of layout than assessing understanding on a particular part of the course, deciding what to learn next, or identifying inaccuracies in the system's model.

In contrast to the detailed conceptual and sequential structures of the predefined views, user-constructed representations were generally more informal and much less detailed, with few students creating their own labels (custom topics), and a minority using links. This is not surprising since the students received no training in techniques such as concept mapping, and Cox [9] notes that representations created for private use tend to be "less fully labelled, sparser, and [sometimes] only partially externalised". Perhaps more surprising is the fact that so few students created hierarchies which were in fact hierarchical, as they may be more familiar with this type of structure due to its everyday use (in file managers etc.). A possible explanation could be that some students are using the self-constructed views as a means of focusing on a portion of the model by including only a small number of topics at a time. With significance attached to presence rather than position of a topic, a seemingly disorganised map or un-nested tree may still be useful. Provision of more generic layout elements (lines, boxes etc.) may enhance support for informal representations.

In some cases students appeared to group topics that did not share any conceptual relationship. It is possible that these layouts would have improved as understanding increased, although results did not indicate a link between ability and correspondence of representation and domain. Nevertheless, layouts which appear meaningless to an observer may still convey meaning to the individual concerned, and constructing an incorrect layout may raise awareness of areas of poor understanding in a way not possible with the predefined views.

Students beginning with the predefined views tended to include more topics in the map (with a larger number of meaningful links) and appeared more likely to use hierarchical relationships in the tree. One explanation could be that users who began with self-constructed views were not aware of how to use links and custom topics, whereas those starting with the predefined views had seen examples. The predefined views may also have improved conceptual understanding, as students using these first were more likely to group conceptually related topics together in their own views. However, students who used the self-constructed views first and made an attempt to organise the topics conceptually but did so erroneously may learn more than students using the predefined views first who simply repeated correct conceptual relationships they observed there.

A longer term evaluation could indicate whether the ability to create accurate conceptual layouts improves as domain knowledge increases, or whether maintaining an accurate layout is not considered important. Involving learners in the analysis phase may provide insight into their intentions when working with the self-constructed views. It would also be useful to evaluate the educational impact of individual views, in order to determine if the inherent representational guidance offered by each can be exploited as a means of focussing attention on particular aspects of understanding (in line with similar work in the area of collaborative discourse [10]). If this is the case, it

may be worth providing training for those students having difficulty using the representations, or including a facility for verifying relationships. There are also possibilities for using learners' self-constructed views to inform the learner model content itself, although if students find ‘inaccurate’ structures helpful as open learner models, a system would need to distinguish whether the learner's created structure actually reflects their understanding. Alternatively, it may be possible to provide the beneficial features of the self-constructed views within the original views. For example, allowing students to hide topics may allow them to focus on a section of the model without needing to create their own view.

3. Summary

Students were given the opportunity to create their own views of an open learner model, and were found to create informal representations lacking in detail, labelling, and in some cases accuracy. There are indications that creating learner model presentations from scratch may pose difficulties for some students, since those using predefined views first used some representational features more fully and often created conceptually more accurate layouts. However, informal representations may still be useful, and the diversity of views created may suggest that learners benefit from the flexibility of creating their own layouts. Further work is needed to assess the effect on learning, and to study how an individual’s representations might change over time.

References

[1] Weber, G. & Specht, M.: User Modeling and Adaptive Navigation Support in WWW-Based Tutoring Systems. *Proceedings of Sixth International Conference on User Modeling,* Springer-Verlag, Wien, New York (1997). 290-300.

[2] Bull, S. & Pain, H.: Did I say what I think I said, and do you agree with me? Inspecting and Questioning the Student Model. *Proceedings of World Conference on Artificial Intelligence in Education*, AACE, Charlottesville VA (1995). 501-508.

[3] Dimitrova, V.: Style-OLM: Interactive Open Learner Modelling, *International Journal of Artificial Intelligence in Education* 13(1), (2003). 35-78.

[4] Zapata-Rivera, J. D. & Greer, J. E.: Exploring Various Guidance Mechanisms to Support Interaction with Inspectable Learner Models. *Intelligent Tutoring Systems: 6th International Conference*, Springer Verlag, Berlin Heidelberg (2002). 442-452.

[5] Ainsworth, S.E, Wood, D.J., and Bibby, P.A.: Co-ordinating Multiple Representations in Computer Based Learning Environments, *Proceedings of EUROAI-ED*, Lisbon (1996). 326-343.

[6] Bull, S. and Mabbott, A.: 20000 Inspections of a Domain-Independent Open Learner Model with Individual and Comparison Views, *International Conference on Intelligent Tutoring Systems*, Springer-Verlag, Berlin Heidelberg (2006). 422-432.

[7] Mabbott, A. and Bull, S.: Student Preferences for Editing, Persuading, and Negotiating the Open Learner Model, 7th *International Conference on Intelligent Tutoring Systems*, Springer-Verlag, Berlin Heidelberg (2006). 481-490.

[8] Papanikolaou, K., Mabbott, A., Bull, S. & Grigoriadou, M. Designing Learner-Controlled Educational Interactions Based on Learning / Cognitive Style and Learner Behaviour, *Interacting with Computers* 18(3). (2006). 356-384.

[9] Cox, R.: Representation Construction, Externalised Cognition and Individual Differences, *Learning and Instruction 9*, (1999). 343-363.

[10] Suthers, D.: Representational Bias as Guidance for Learning Interactions: A Research Agenda. *Artificial Intelligence in Education,* IOS Press, Amsterdam (1999). 121-128.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Translation of Overlay Models of Student Knowledge for Relative Domains Based on Domain Ontology Mapping

Sergey Sosnovsky[1], Peter Dolog[2], Nicola Henze[3], Peter Brusilovsky[1], Wolfgang Nejdl[3]
[1]*School of Information Sciences, University of Pittsburgh, USA*
[2]*Department of Computer Science, Information Systems Unit, Aalborg University, Denmark*
[3]*L3S Research Center, University of Hannover, Germany*
sas15@pitt.edu, dolog@cs.aau.dk, henze@l3s.de, peterb@pitt.edu, nejdl@l3s.de

Abstract. The effectiveness of an adaptive educational system in many respects depends on the precision of modeling assumptions it makes about a student. One of the well-known challenges in student modeling is to adequately assess the initial level of student's knowledge when s/he starts working with a system. Sometimes potentially handful data are available as a part of user model from a system used by the student before. The usage of external user modeling information is troublesome because of differences in system architecture, knowledge representation, modeling constraints, etc. In this paper, we argue that the implementation of underlying knowledge models in a sharable format, as domain ontologies - along with application of automatic ontology mapping techniques for model alignment - can help to overcome the "new-user" problem and will greatly widen opportunities for student model translation. Moreover, it then becomes possible for systems from relevant domains to rely on knowledge transfer and reuse those portions of the student models that are related to overlapping concepts.

Introduction

Adaptive educational systems (AES) have a long history of research studies that can be traced back to early 70-s [1]. Over the years, various types of AESs [2] have demonstrated their effectiveness. The central component of every AES is a student model (SM), where the system stores its beliefs about the different characteristics of a student. The proximity of modeling assumptions to the actual state of a student determines how adequate the system's adaptive interventions can be. One of the problems shared by all adaptive systems is the "*new-user"* problem. The models of those students who have just started their work with the system are inherently poor, since the system has little or no evidence from which to draw its assumptions.

There are three traditional approaches to dealing with the "new-user" problem in modeling student knowledge. The majority of the systems use the so-called *tabula-rasa* (clean slate) approach, i.e., they assume that the user has no knowledge of the subject. Some systems begin, instead, with a reasonably sized questionnaire, aimed at assessing the user's starting level of knowledge and thus seed the user model accordingly, e.g., ELM-ART [3]. Systems supporting editable student modeling allow students to enter their knowledge levels based on intuition and previous domain experience [4]. Neither of these approaches is perfect. The first one is not appropriate in situations where new users are likely to have some previous reasonable knowledge of the subject. The second consumes time and intervenes with the educational process. Finally, editable SMs rely on subjective opinion of a student, which might be influenced by irresponsible or biased judgment and inability to correctly assess own knowledge.

Our paper suggests an alternative approach: to initialize the SM of one AES based on an existing model of this student built by another AES. This approach avoids time-consuming questionnaires and potentially biased self-evaluation while attempting to achieve a good quality of student modeling from the very beginning. Such scenarios have become quite realistic nowadays. As the WWW has become the ultimate platform for information and service delivery, numerous Web-based AESs have appeared [2]; hence it is possible to expect that the same student would interact with several AESs during his/her course of study. Translation (or mediation) of user models from one system to another is an emerging research field of user modeling (see for example, [5], [6]). Unfortunately, the design of most AESs does not support model sharing. Several teams are working on the development of distributed architectures for online learning, where AESs operate as components, using the same central UM server, e.g., ActiveMath [7], Personis [8]. However, these

systems cannot be easily plugged into a different architecture because of the discrepancies in communication protocols and model representation.

The problem of SM mediation becomes even harder if we consider mediation beyond a particular domain, e.g., between C and Java programming. Such domains share a large number of concepts; therefore, for a student studying C programming, a knowledge transfer [9] to Java should occur. Let's imagine that our student has been studying C programming with the help of a standalone AES, which collected an overlay model [10] of his/her knowledge. Now the student is planning to use another system for learning Java. In this situation the *tabula-rasa* SM of the AES teaching Java would not reflect the actual state of knowledge s/he has. As a result, the adaptation performed by the Java-system will not be adequate; it can even be potentially harmful. We could avoid this by using information from the SM of the C-system to initialize relevant parts of the SM representing Java knowledge (see fig. 1). However, first, it would be necessary to translate this information so that the second system would be capable of understanding it.

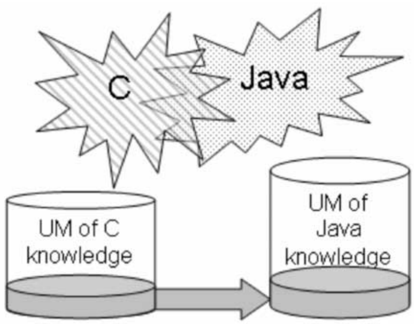


Figure 1. The relationship between C and Java knowledge.

We propose a solution to the described problem, based on the technologies that have recently become available in the framework of the Semantic Web initiative [11]. The implementation of the AES's domain model in a sharable format, as a web-ontology, enables access to this model by other AES. The ontology mapping techniques [12] help to automatically identify similar concepts in the ontologies of related domains and align the domain models of the two AES. Once found, the mapping can be used as a template for the translation of overlay SMs from one system to another. The idea to apply ontologies for knowledge representation in AESs is not new [13], [14]. Several attempts have been made towards implementation of ontology-based interoperable SMs. Denaux et al. [15] developed Onto-AIMS, a system integrating two AESs using a shared ontology. Authors of [16] describe Medea, a learning portal supporting cross-application adaptation on the basis of manual mapping of ontologies underlying a particular AES to the central domain ontology. Unlike these projects our approach does not imply a specific ontology shared my multiple AESs or a core system, used as a central interpreter by others. We investigate the possibility of using general ontology mapping techniques to *automatically* translate between SMs based on *arbitrary* ontologies from *relative* domains sharing a set of concepts. To validate our hypotheses, we have conducted an experiment in an introductory Java class. The statistical analysis demonstrates that automatic ontology mapping provides a solid basis for translation of overlay SMs between relative domains.

The rest of the paper is structured as follows. Section 1 describes the ontologies and educational content models, as well as gives the details of the mapping algorithm. Section 2 analyzes the results of the conducted experiment. Section 3 presents further discussion questions and sketches the directions of future work. Finally, section 4 concludes the paper.

1. Ontology Mapping for Translation of Student Models

We assume here that the systems whose domain models are being aligned follow basic principles of the formal representation of domain knowledge by ontologies. However, even when the AESs operate on similar domains, their domain models usually differ in structure, concept labels, granularity and modeling focus, as well as in the indexing and instantiation of educational content. Therefore, as a basis for ontology mapping, we chose an algorithm which assesses concept similarity based on their instances, using machine learning (ML) techniques. Such algorithms seem to be more robust towards managing the modeling discrepancies listed above. Instead of the intended meaning of the concepts being mapped (*what is the concept?*) they concentrate

on the operational meaning (*what is the concept about?*). The input data for any ML-based ontology mapping algorithm are the two ontologies, along with the sets of instances corresponding to the concepts of these ontologies. In the framework of AES, an instance of a concept is an occurrence of this concept in a learning resource. The next subsection describes the ontologies we mapped, along with their accompanying instances.

1.1. Domain Ontologies and Content Modeling

In our experiment we used two existing ontologies – one for Java and one for C. The Java Ontology was developed for the AES *Personal Reader for eLearning* [17]. The ontology is represented in RDFS and can be accessed at *http://personal-reader.de/rdf/java_ontology.rdf*. It defines 490 *rdfs:Class*'s connected to each other by the generic relation *rdfs:subClassOf*. The Java Ontology has been used to index the online SUN Java Tutorial (*http://72.5.124.55/docs/books/tutorial/java/index.html*), which primarily has been used as a resource for developing this ontology. The description of the tutorial structure and the index of its pages in terms of the concepts of the Java Ontology is represented as another RDF document [17].

The C Programming Ontology was also developed in RDFS. It was designed at the University of Pittsburgh and can be accessed at *http://www.sis.pitt.edu/~paws/ont/c_programming.rdfs*. The ontology defines 575 *rdfs:Class*'s and 5 *rdf:Property*'s. Table 1 summarizes some basic characteristics of the two ontologies. The Java Ontology covers a much bigger domain (including knowledge of object-oriented programming and some parts of the Java standard library); therefore the percentage of manually-mapped concepts is much smaller for Java than it is for C. At the same time, the C Programming Ontology provides a finer-grained representation of the domain's conceptual structure. During the manual mapping, we found many situations where one Java concept mapped to several C concepts; for example, the manual mapping contains more C concepts than Java concepts.

Table 1. Basic metrics of the two ontologies.

	Java Ontology	**C Programming Ontology**
Number of rdfs:Class's (concepts) defined	490	575
Number of concepts mapped	108 (22%)	257 (45%)
Number of rdf:Property's defined	0 (uses rdfs:subClassOf)	5 (isA, partOf, directPartOf, implement, utilize)
Number of RDF-triples	1013	1538

The focus of modeling for the C Programming Ontology was also different. While the Java Ontology was initially created to describe the content of the SUN tutorial, the C Programming Ontology was deliberately developed as a general-purpose ontology, in an attempt to model the domain of C programming in a maximally precise and unbiased way. These differences in modeling perspectives added additional difficulties for mapping. Figure 2 visualizes an extract from the C Programming Ontology. The content for indexing by the C Programming Ontology was taken from two online C tutorials: University of Leicester C Tutorial (*http://www.le.ac.uk/cc/tutorials/c/*) and Rob Miles's Tutorial (*http://www.eumus.edu.uy/eme/c/c-introduction_miles/contents.html*).

1.2. Ontology Mapping Algorithm

To perform ontology mapping, we implemented a modified version of the GLUE approach [18]. This algorithm computes the matrix of similarities between all concepts in two ontologies. The main idea of GLUE is based on the cross-classification of concept instances, counting of instances associated with concepts from different ontologies and following the calculation of joint probabilities for each pair of concepts. GLUE was chosen for its clear and robust algorithm, which could be easily adjusted for our purposes. A tutorial page indexed with a concept is seen as its instance. Pages were classified based on their term vectors. Term vector calculation included stemming (with the help of Porter stemmer), stop word filtering and TFIDF computation.

The quality of the GLUE algorithm is determined by the precision of the classifiers, which depends highly on the quality and size of the training data sets. To reduce this dependency, we added an extra mapping step based on the concept names. In this step, concept names are now parsed into separate words and compared based on lexical similarity. To calculate the lexical similarity of two words, we used Levenshtein's edit distance [19] and the metric proposed in [20]. The output of name-based mapping is used as the input for instance-based mapping.

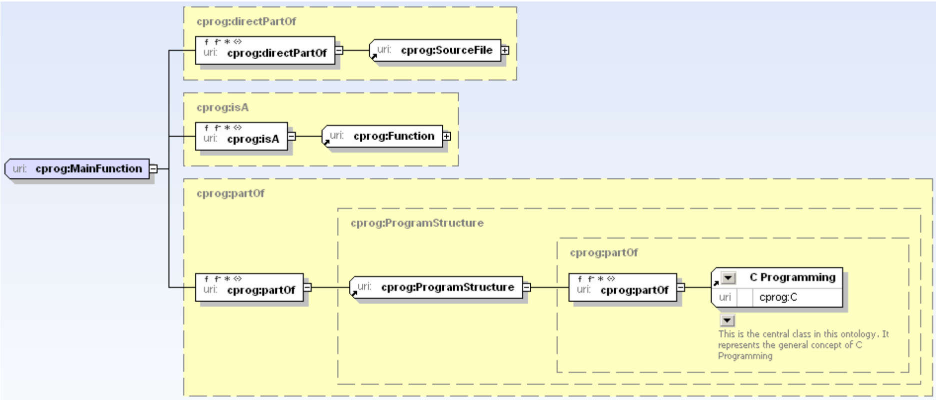


Figure 2. An extract from the C Programming Ontology.

Another characteristic of the input data, which has had a negative impact on the mapping precision, was the large percentage of instances shared by several concepts (i.e., tutorial pages are indexed with a number of concepts). This situation is typical for adaptive content authoring, but not for ontological engineering. As a result, the mapping algorithm could not distinguish between close concepts that often occur together (e.g. *StatementIf* and *RelationalOperator*). To deal with this problem, when training a classifier for a concept, the mapping algorithm weights term vectors according to the degree of influence that the concept has on this tutorial page. Hence a page indexed with a set of *n* concepts, including the concept *A,* contributes to the training of the classifier for this concept *n* times less then a page whose index contains only the concept *A*.

The original GLUE algorithm was developed to find the best possible "1-to-1" mappings for all concepts, from the two ontologies. In our algorithm, we relaxed this requirement, since our main goal was not the ontology mapping itself, but student knowledge translation based on the found mapping of domain ontologies. Sometimes the mapping algorithm is unable to distinguish between closely-related concepts because they have similar names and definitions, as well as share many instances. However, this is less important in the context of knowledge modeling, because students tend to have similar knowledge for related concepts (e.g., *WhileLoop* and *DoWhileLoop*). Hence, instead of finding the best "1-to-1" mapping, our algorithm finds several best candidates for every concept, where mapping is possible. These candidates are weighted according to the similarity values found during the mapping process. When SMs are translated these weights determine the percentage of contribution by candidate concepts from the first ontology to the resulting knowledge level of the target concept from the second ontology. This feature helps deal with the "1-to-many" mappings, especially when the candidate concepts are from different parts of the ontology. For example, good candidates for the concept *return* from the Java Ontology would be concepts *FunctionReturnValue* and *StatementReturn*, which belong to different branches of the C Programming Ontology.

2. The Experiment

2.1. Data Collection

The experiment was conducted in an introductory course on Object-oriented programming, which was being taught to undergraduate students in the School of Information Sciences, at the University of Pittsburgh. At the beginning of the course, we collected a questionnaire assessing their initial experience in C and Java programming. Out of 36 students, seven demonstrated a prior knowledge of Java and therefore were not eligible for the main experiment (we will call this group the "*Java"-group*). Eight students knew neither C nor Java; they formed the control group for the evaluation of C-model quality (we call it the *"¬C ¬Java"-group*). Finally, 21 students had C programming experience but had never used Java before. They became our main experimental group (*"C ¬Java"-group*). During the analysis of outliers, six cases were filtered out: three by their Java test scores and three by their C test scores. The final number of students in the groups was respectively 6, 7 and 17.

The 10-question pre-quiz was used for the initialization of the SM for C programming. As a result, we estimated the prior knowledge of 37 concepts from our C ontology. To evaluate Java knowledge, we used weekly quizzes. Each of these quizzes included 2-3 extra-credit questions checking the student's knowledge of several

Java concepts that have some analogy in C language. At the time of the quiz these concepts were not yet mastered in class, so the recorded knowledge of these concepts was a result of transfer from C, not the newly acquired knowledge. This gave us overlay models of the knowledge of 21 concepts of the Java Ontology. The problems (both in C pre-quiz and in Java extra-credit questions) assessed the students' knowledge on the level of application – they had to analyze code fragments and answer related questions.

To create the "*ground truth*" for our automatic ontology mapping experiment, we performed the manual mapping of these sets and identified 17 mapping cases from four large categories: *Functions/Methods*, *Operators*, *Simple Data Types* and *Control Structures*. All 17 cases were "1-to-1" mappings. They were part of the best mappings one could achieve between these two ontologies, for which we had overlay models of student knowledge in both C and Java. The automatic ontology mapping reduced this subset to nine concepts. It became our target set of concepts. For these nine concepts we were able to compare the results of the SM translation, based on the automatic ontology mapping with the translation being based on the manual ("*perfect*") mapping.

2.2. SM Validation

The results of the questionnaire on programming experience (the size of the *"C ¬Java"-group*) confirmed the scenario described in the introduction, i.e., the presence of a sufficient number of students moving from C to Java. However, the problem for our experiment was to find enough students of that kind *with the populated SM of C knowledge*. This goal was hard to accomplish in the scale necessary for statistical data analysis. Therefore, the C-knowledge SMs have been initialized using the pre-quiz. To verify the results of the pre-quiz and consequently to validate collected SMs, we used the data from the programming experience questionnaire.

The population of students who did not know Java consisted of both the *"¬C ¬Java"-group* (7 subjects) and the *"C ¬Java"-group* (17 subjects). For all students, we calculated the cumulative level of C knowledge based on the results they showed on the pre-quiz for the target set of concepts. The necessary assumptions of parametric statistical tests were not met. The distributions of data across both groups were severely skewed from the normal curve and the data sample variances were not homogeneous as well. Therefore, to compare the mean scores of these groups we employed the Wilcoxon-Mann-Whitney test as a t-test substitute. The mean score for students with C experience (*0.94 ± 0.01*) was significantly greater then the mean knowledge level for students who were novices in C programming (*0.64±0.11*): *Mann-Whitney U statistics = 19.00; p = 0.010*. Table 2 summarizes the results of statistical analysis.

Similarly, we estimated the validity of the Java SMs. The average score for the mapped subset of Java concepts was calculated for all students. The entire population of students was divided into three groups: the *"¬C ¬Java"-group* (*mean score = 0.82±0.04*), the *"C ¬Java"-group* (*mean score = 0.93±0.02*) and the "*Java"-group* (*0.98±0.01*). The data distribution in two groups was skewed from the normal and the variance of data across groups was not homogenous. Therefore, we used a non-parametric statistical test as a substitute for the one-way ANOVA. The Kruskal-Wallis test has demonstrated that there is a statistically significant difference in the cumulative level of Java knowledge among students from different groups: *Chi-Square statistics = 7.408; p = 0.006*.

Before saying that the automatic mapping of one model into another is possible and beneficial, it is worthwhile to validate the underlying assumption that the knowledge transfer from C to Java did happen. As a target set here we used students from both the *"¬C ¬Java"-group* and the *"C ¬Java"-group*. We did not use the third group of students, since their performance on the Java pre-quiz depended directly on their previous Java knowledge, rather than on the knowledge of related C concepts. To estimate the strength of the relationship between the knowledge of the target sets of C and Java concepts, we applied a correlation analysis. The Pearson correlation could not be used, since our dataset did not meet a number of underlying statistical assumptions. However, the non-parametric Kendall's correlation analysis demonstrated a statistically significant relationship between the results on the C and Java pre-quizzes: *Kendall's Tau b = 0.294; p = 0.050*. Since students from the analyzed group did not have any previous knowledge of Java before the course, we can credit such a relationship to the transfer of knowledge from C to Java.

2.3. SM Translation Based on Automatic Ontology Mapping

To evaluate the quality of the SM translation based on automatic ontology mapping we compared it with "*ground truth*," as provided by the results of manual SM translation. We represented SMs as vectors, where every concept is a coordinate. As a result, the student knowledge of nine related C and Java concepts are characterized by two 9-dimensional vectors (as an example, fig. 3 shows the SM of knowledge of the C concepts *StatementFor* and *StatementIfElse* and the SM of knowledge of the corresponding Java concepts *for* and *else*, represented as 2-dimensional vectors). Since the manual mapping we performed was the best possible one-to-one mapping, we can

say that the dimensions of the C and Java vectors are collinear. Consequently, we can superimpose these coordinate spaces and calculate the Euclidian distance between two vectors, which reflects the divergence between the initial C and Java knowledge of the student.

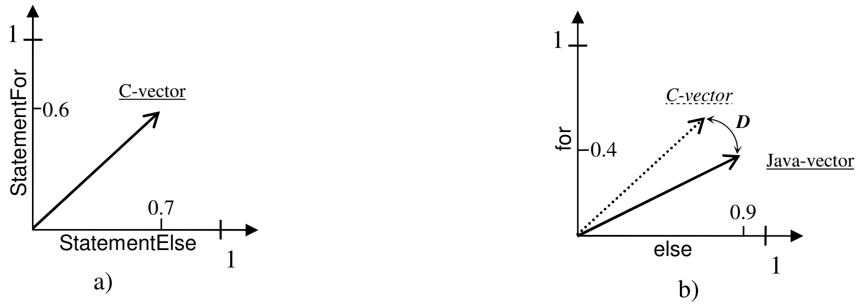


Figure 3: The SM of knowledge for two concepts represented as a 2-dimensional vector.
a) The C-model (knowledge level of *StatementIfElse* is 0.7; knowledge level of *StatementFor* is 0.6).
b) The Java-model (knowledge level of *else* is 0.9; the knowledge level of *for* is 0.2) with the C-model superimposed. Corresponding concepts of the C and Java models are mapped to each other as 1-to-1; therefore the dimensions of the two coordinate spaces are collinear. Euclidian distance ***D*** shows the divergence between the C-model and the Java-model.

Based on the manual mapping as well as the results of the C pre-quiz and Java extra-credit questions, we calculated the divergence distances for all students from the experimental group (*"C ⌐Java"-group*). These distances characterize both the shortcomings of the student modeling, due to the simplification of modeling assumptions and defects in elicitation process, as well as possible differences in the students' understanding of the relevant concepts of C and Java, unexpected mistakes (or guesses) on pre-quizzes, etc. Overall, these distances characterize the intrinsic imperfection of the obtained dataset. It is not possible to compute a set of smaller distances, i.e., to perform a better translation of the SMs on the basis of the given data.

Automatic mapping provided us with another set of distances calculated only on the basis of C knowledge. These distances characterize the differences between the original Java model vectors obtained from the Java pre-quiz and the Java model vectors computed by translating the SMs of C programming. Every coordinate (every concept's level of knowledge) of later vectors is calculated based on the results of automatic ontology mapping and the knowledge of C concepts being mapped. To verify that the distances obtained manually are close enough to the estimates based on the automatic mapping, we applied correlation analysis. The results (*Kendall's Tau b = 0.844; p << 0.05*) demonstrate an exceptionally strong relationship between the two variables.

Hence, we have shown that the SMs of Java knowledge obtained with the help of automatic translation from the SMs of C knowledge are very close to those computed manually, i.e., automatic mapping provides close to "perfect" translation of knowledge levels from the C to Java domains.

Table 2. Results of the statistical analysis.

Goal	Type of analysis	Results
Validation of SM of C knowledge	Wilcoxon-Mann-Whitney test for 2 independent samples	Mann-Whitney U statistics = 19.0 (**p = 0.010**)
Validation of SM of Java knowledge	Kruskal-Wallis test for k independent samples	Chi-Square statistics = 7.408 (**p = 0.006**)
Relationship between C and Java knowledge	Kendall's test of correlation	Kendall's Tau b = 0.294 (**p = 0.050**)
Relationship between SM translations based on manual and automatic mapping	Kendall's test of correlation	Kendall's Tau b = 0.844 (**p << 0.05**)

3. Discussion and Future Work

The described experiment was performed for two related but different domains; however, the solution we propose could also be applied for translating SMs collected by different systems in the same domain. Two knowledge

engineers would likely model the same domain in different ways. The modeling information is too precious; modern AESs need to be capable of translating each others' SMs.

On the other end of the scale lies the situation where the domains are fairly different, such as Biology and Geography. Can knowledge mediation in such domains still benefit by ontology mapping (probably for the very high-level concepts)?

A more theoretical question is: what are some general recommendations for close-domain ontology mapping? When is it worthwhile? What are the metrics of domain closeness? How can we say whether model translation is possible for two domains? It could be the percentage of related concepts. It could be the closeness of domains of interest in the general hierarchy of domains based on some common sense upper-level ontology, such as SUMO [21], DOLCHE [22] or CYC [23]. It could be some metric from the field of information retrieval, obtained by the analysis of related recourses.

The work described in the paper is only a first step. There are several possible directions of how we might continue this project:

- Differences in domain model representation are not the only source of discrepancy between overlay SMs. The scales used to assess knowledge levels should be mapped as well. They might be categorical vs. continuous; lower and upper borders could differ, as well as scale intervals. The formula for knowledge level calculation might also make a difference. For example, the same numeric value on the fixed scale calculated as a weighted average progress, or as a logarithm of the number of correct attempts, or as a sigmoid function value can represent different knowledge levels.
- Although the results of the mapping algorithm are satisfying, we could further improve its performance and precision. The current version on our server maps the C and Java ontologies used in about 30 minutes; such performance does not allow the building of an interactive application. It also needs to adjust to the possible extension of data sets and/or bigger ontologies. The utilization of structural relations between concepts as well as the development of a more elaborate, name-based mapping component should improve the precision of the algorithm. For example, we might use WordNet [24] for synonym processing or take into account the TFIDF values of different words in the concept names, in order to extract meaningful words from names like *StatementWhile, StatementIf, StatementFor*, etc.
- It is interesting to test the proposed approach in different domains. What would be the results for two different conceptualizations of the same domain? Or for domains less related then C and Java?
- Finally, it would be a good idea to develop an automated community service performing SM translation online.

4. Conclusions

In this paper, we investigated two main research questions:

- Is it possible to initialize a SM of an AES using another AES's SM acquired in a related but different domain?
- Is ontology mapping an appropriate mechanism for doing this?

The analysis of experimental data shows that automatic ontology mapping provides a sufficient basis for SM translation from one domain to another. Certainly, the obtained values are not perfect; however, they provide a better estimation of the starting levels of knowledge than the dominant *tabula-rasa* approach. The main advantages of the proposed approach are as follows:

- It is **fully automatic**: Once found, the mapping could be used for translating any number of SMs from one ontology to another.
- It is **bidirectional**: The found mapping could be used for the translation of SMs in both directions (for example, in the analyzed scenario: from both C to Java and Java to C).
- It is **instructionally-safe**: The translation process does not interfere with the learning process; students are not required to pass additional tests or to estimate their own knowledge.
- It is **objective**: Its performance is defined by the granularity and modeling constraints of the domain ontologies, as well as the accuracy of the mapping algorithm; it does not depend on any subjective decision made by a student or an instructor.
- It is **domain independent**: As long as two ontologies share some set of concepts, and mapping occurs, translation is possible.

We strongly believe that application of ontological technologies for student modeling can help to overcome a number of AES problems. The reported project is a step in this direction.

References

[1] Carbonell, J.R., AI in CAI: An Artificial Intelligence Approach to Computer-Assisted Instruction. IEEE Transactions on Man-Machine Systems, 1970. 11(4): p. 190-202.

[2] Brusilovsky, P. and C. Peylo, Adaptive and intelligent Web-based educational systems. International Journal of Artificial Intelligence in Education, 2003. 13(2-4): p. 159-172.

[3] Weber, G. and P. Brusilovsky, ELM-ART: An adaptive versatile system for Web-based instruction. International Journal of Artificial Intelligence in Education, 2001. 12(4): p. 351-384.

[4] Mabbott, A. & Bull, S., Student Preferences for Editing, Persuading and Negotiating the Open Learner Model, in M. Ikeda, K. Ashley & T-W. Chan (eds), Intelligent Tutoring Systems: 8th International Conference, Springer-Verlag, Berlin Heidelberg: p. 481-490.

[5] Berkovsky, S., T. Kuflik, and F. Ricci. Cross-Technique Mediation of User Models. in 4th International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH'2006). 2006. Dublin, Ireland: Springer Verlag: p. 21-31.

[6] Carmagnola, F. and F. Cena. From Interoperable User Models to Interoperable User Modeling. in 4th International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH'2006). 2006. Dublin, Ireland: Springer Verlag: p. 409-414.

[7] Melis, E., et al., Semantic-Aware Components and Services of ActiveMath. British Journal of Educational Technology, 2006. 37(3): p. 405-423.

[8] Kay, J., R.J. Kummerfeld, and P. Lauder. Personis: a server for user models. AH'2002, Adaptive Hypertext 2002. 2002: Springer: p. 203-212.

[9] Singley, M.K. and J.R. Anderson, The transfer of cognitive skill. 1989, Cambridge, MA: Harvard University Press.

[10] Carr, B. and I. Goldstein, Overlays: A Theory of Modelling for Computer Aided Instruction. 1977, Massachusetts Institute of Technology, Cambridge AI Lab. p. 24.

[11] W3C. Semantic Web. 2001. Available from: http://www.w3.org/2001/sw/.

[12] Kalfoglou, Y. and M. Schorelmmer, Ontology mapping: the state of the art. The Knowledge Engineering Review, 2003. 18(1): p. 1-31.

[13] Mizoguchi, R., Sinitsa, K. and Ikeda, M., Task Ontology Design for Intelligent Educational/Training Systems. Workshop on Architectures and Methods for Designing Cost-Effective and Reusable ITSs in conjunction with ITS'96, Montreal, 1996: p. 1-21.

[14] Mizoguchi, R. and Bourdeau, J., Using Ontological Engineering to Overcome AI-ED Problems, International Journal of Artificial Intelligence in Education, Vol.11, No.2, 2000: p.107-121.

[15] Denaux, R., Dimitrova, V., Aroyo L., Integrating Open User Modeling and Learning Content Management for the Semantic Web, User Modeling 2005: 10th International Conference, Edinburgh, Scotland, UK. Springer, 2005: p. 9-18.

[16] Trella, M., C. Carmona, and R. Conejo. MEDEA: an Open Service-Based Learning Platform for Developing Intelligent Educational Systems for the Web. Workshop on Adaptive Systems for Web-Based Education: tools and reusability (AIED'05). 2005. Amsterdam, the Netherlands: p. 27-34.

[17] Henze, N., P. Dolog, and W. Nejdl, Reasoning and Ontologies for Personalized e-Learning in the Semantic Web. Educational Technology & Society, 2004. 7(4): p. 82-97.

[18] Doan, A., et al., Learning to map between ontologies on the semantic web. 11th International World Wide Web Conference. 2002. New York, NY, USA: ACM Press: p. 662-673.

[19] Levenshtein, I.V., Binary Codes capable of correcting deletions, insertions, and reversals. Cybernetics and Control Theory, 1966. 10(8): p. 707-710.

[20] Maedche, A. and S. Staab. Measuring Similarity between Ontologies. 13th International Conference on Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web 2002: Springer-Verlag.

[21] Niles, I. and A. Pease. Towards a standard upper ontology. FOIS '01: Proceedings of the international conference on Formal Ontology in Information Systems. 2001. Ogunquit, Maine, USA: ACM Press.

[22] Gangemi, A., et al. Sweetening Ontologies with DOLCE. Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web, 13th International Conference, EKAW 2002. 2002. Siguenza, Spain: Springer Verlag: p. 166-181.

[23] Lenat, D.B., CYC: a large-scale investment in knowledge infrastructure. Communications of ACM, ACM Press, New York, NY, USA, 1995. 38(11): p. 33-38.

[24] Miller, G.A., et al., Introduction to WordNet: An On-line Lexical Database. International Journal of Lexicography, Oxford University Press 1990. 3(4): p. 235-244.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Bootstrapping Accurate Learner Models from Electronic Portfolios

Zinan GUO and Jim GREER
ARIES Laboratory, Department of Computer Science, University of Saskatchewan
Saskatoon, Saskatchewan, S7N 5C9, CANADA
zig094@mail.usask.ca, jim.greer@usask.ca

Abstract: This paper presents a way to use electronic portfolios (e-portfolios) to initialize learner models for adaptive learning environments. E-portfolios become sources of evidence for claims about prior conceptual knowledge or skills. We developed the EP-LM system to test whether accurate learner models can be built from e-portfolios and to determine whether this approach can be beneficial for reflective learning. EP-LM supports e-portfolio browsing, claiming skill levels on certain concepts, and evidencing the claims using artifacts included in the e-portfolio. Experimental results show that accurate learner models can be created through this approach and that reflective and personalized learning can result.

Keywords: electronic portfolio, learner model, intelligent tutoring system.

1. Introduction

An e-portfolio is defined as an organized collection of digital and/or analog artifacts and reflective statements that demonstrate a learner's intellectual development over time. [1] The rapidly growing use of e-portfolios in higher education provides students a user-centered file management option. Many schools and universities have developed e-portfolio systems where students are encouraged to store and organize artifacts during their formal schooling and to further carry on with augmenting that e-portfolio during lifelong learning. E-portfolios can offer advantages in demonstration of skills, learner reflection, collaboration, and assessment. The portfolio offers the possibility to show how the learner *conquers* the subject domain during work, study or other learning process. In this lies also the possibility to discover connections across subject areas and to develop a meta-perspective on the student's own learning and knowledge [5]. E-portfolio assessment involves evaluating learner achievement relative to pre-defined rubrics and utilizing the e-portfolio elements as evidence sources for achievement of prescribed learning outcomes. This has some striking similarity to the method described in this paper for initializing learner models. It also bears some similarity to inspectable learner modelling. The use of e-portfolios allows qualifications to be changed to focus more on the actual core of what needs to be assessed, rather than peripheral efforts to record and administer assessment tests of students' work [2].

Learner (or student) models are specialized mechanisms for representing information and knowledge of learners inside adaptive learning environments. In general, learner models are in the form of detailed cognitive models and are maintained along with learning interactions in order to recommend suitable learning resources or to

provide individualized help during problem solving. Open, inspectable student models help learning by enabling improved learner reflection [9]. Research about creating inspectable student models to support learner reflection by exploiting evidence from student use of web-based learning resources shows promising results [8]. In this paper, we analyze the interoperability between e-portfolios and learner models in university-level educational settings.

2. Create Sample E-Portfolios

A standardized e-portfolio information model explains what information is generally contained in e-portfolios, while a set of specifications are defined to describe the organization of the e-portfolio [7]. The IMS ePortfolio information model specification has been proposed and deemed plausible. With this specification, standardized meta-data can be bound to artifacts in an e-portfolio making them easier to interpret and maintain across different institutions and platforms/systems [4]. This standard structure makes it easy for developers and users to query and track portfolio parts at different granularity levels and reference them in other applications by making reference simply to a universal resource identifier.

The sample e-portfolios created for this research are course e-portfolios that can be included as part of the learner's life-long learning portfolio. The sample e-portfolios contain all the teaching and learning materials accessed and produced by six different student volunteers in an entry-level Java programming course taught over the summer of 2006. The course provides a broad introduction to fundamental concepts of computing, object oriented computer programming, and how to write programs in Java. We collected and organized the artifacts in the following categories:

- Lecture notes, tutorials and lab documents
- Assessment, which consists of 3 smaller and 3 larger assignments
- Five quizzes, one midterm exam, one lab exam and the final written exam.
- Students' posts and activities from the iHelp discussion forums.

Some of the artifacts could be imported from existing learner support systems such as the electronic assignments submission system, but others were paper-based and needed to be converted into electronic format. To keep it simple and easy to control, we chose to build the e-portfolios from scratch instead of using any off-the-shelf e-portfolio system. We created and included meta-data describing some important ontological relationships among artifacts according to the course syllabus/concept map. At the end of the term, we interviewed the six student volunteers in order to let them evaluate the e-portfolios, to collect self-reflective comments on different artifacts, and to fill out a questionnaire claiming their knowledge levels on various concepts. The interview results showed that all six students believed that their e-portfolio covers/represents their knowledge and they were willing to share their e-portfolios with future instructors. However, one expressed some concern about privacy.

3. Initializing Learner Models

Traditionally, learner models are created either by making default guesses about the learner or by having learners complete some diagnostic questionnaire through which

they demonstrate their initial knowledge levels. The accuracy is strongly affected by domain/subject matter, and tends to be hard to verify. Extracting information as learning evidence from e-portfolios to initialize student models will enrich the content of learner models and may potentially increase the learner model's accuracy. Learner models generated by our system include not only the student's knowledge level on each question/concept shown in Table 1 (e.g. How capable is the student with <concept i>?), but also lists selected artifacts that may support the claims on each concept.

Table 1: Main concept that contribute to the learner model for an "Abstract Data Type" tutoring system

1	Define Variables/Methods/Classes	6	Nested Object
2	Method parameters/Return statement	7	Complex object (class with multiple data types)
3	Constructors	8	Simple Arrays/Vectors
4	Control Structures	9	Search and sort array/vector
5	Object Concept	10	Concept of Abstract Data Type

Three main methods of initializing/bootstrapping a learner model are: 1. users outlining their own learning goals; 2. users providing a self-description (general personal information); 3. users being given a pre-test on the subject area [3]. Both 1 and 2 are easy to adapt to e-portfolio information extraction with partial automation and assistance, but achieving 3 requires fairly detailed e-portfolio instances that need to be mapped and interpreted from a standardized information model. For this reason, we focus on an approach where a customized interface is provided for permitting students (or teachers) to link evidence from an e-portfolio with answers to reflective questions about the learner's knowledge-state. Previous research has shown reflective comments (from human tutors and peers) about problem-solving activities to be effective in helping students to reason about their own learning behaviours [6]. Since some students may have a hard time remembering details about their prior learning activities, this evidence-linking ("evidencing") process is believed to be helpful as it not only "forces" students to browse their own learning records, but provides assistance in e-portfolio browsing and learner model editing. We believe that the evidencing process provides benefits in three aspects of learning: 1. Richer content can be expressed: prior learning experiences can be mapped to skill levels as supporting evidences in the learner models. 2. Reflective learning can occur as students get a chance to review their prior learning via assisted functions provided by the system. 3. The approach promotes a portfolio-based assessment model.

4. The EP-LM System

EP-LM is a system to initialize learner models from e-portfolios. This is accomplished by making claims about the skill level on various concepts and backing up the claim with evidence drawn from the e-portfolio. The content and associated meta-data in an e-portfolio serve as the input. Although e-portfolios do not represent explicit models of cognitive capabilities, they contain evidence that may justify claims about the learner's knowledge or skills. The extension to the learner model is essentially a set of questions and an interface to e-portfolios. The questions for bootstrapping the learner model are defined based on the specific requirements of the adaptive learning environment (ALE). Evidencing is the process of linking e-portfolio artifacts to claims about knowledge or

skills. For each concept, the user needs to specify a knowledge level or make a claim. Some artifacts are highlighted because they are deemed more relevant to the current question according to the course syllabus/ontology.

EP-LM provides an interface for performing the "evidencing" and "e-portfolio browsing" functions. For each concept, the user is asked to specify a knowledge level (None, Poor, Average, Good and Excellent) by choosing an item in a group of radio buttons. The e-portfolio artifacts are listed in a table at the bottom of the evidencing page, and can be view by clicking on the name of an artifact. The clicked artifact will be shown in a pop-up window, where this artifact can be added to evidence by clicking the *Add to Evidence* button in the header of the pop-up. The user can specify a support type and comment when adding an artifact to the evidence list. The pop-up window also provides an option for the user to be able to add the viewed artifact as evidence to other concepts by changing the *Evidence For* drop-down list.

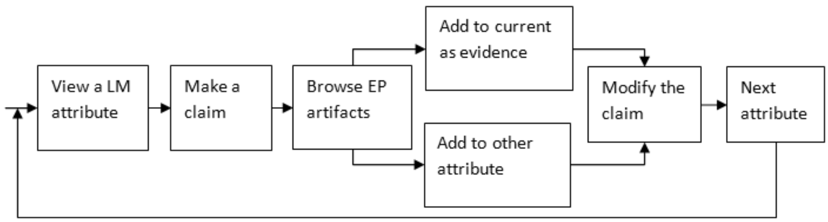


Figure 1: User activities in the EP-LM system

5. The Experiment and Results

The purpose of the experiment is to evaluate how accurately learner models can be created through this approach and to discover how beneficial such an approach can be in terms of reflective learning. With six student volunteers we constructed authentic e-portfolios that documented their learning in an introductory Java programming course. We then selected four of the e-portfolios that covered the full range of student achievement in the course. We invited five domain experts to participate in the experiment, each of whom went through four e-portfolios and proceeded through the “evidencing” process. The experts were also interviewed together where they had some discussions about their judgments and comments on the approach and results.

Each expert created four learner models using the four student e-portfolios. Experts made estimates of each student’s knowledge level on the ten concepts shown in Table 1. They also browsed the e-portfolios of the students and linked artifacts as evidence in supporting their estimates using EP-LM. An example of the data collected as one expert browsed one student e-portfolio to construct estimates of student knowledge levels is presented in Table 2.

5.1. Evaluate Data Consistency and Reliability

We examined the inter-rater reliability among the experts’ estimates of knowledge level of each learner on each concept. The Cronbach’s alpha statistic was 0.809 with 95% confidence interval, which indicates a “good” level of agreement according to [10] and [11]. Looking more closely at Table 2, there was a large standard deviation

among experts on concept 6, the *nested object concept*. In debriefing interviews with the experts it was revealed that there was some variation in understanding of the intent behind this concept by the experts. The concept was not explicitly taught in the prior course, but some experts judged learners' readiness to easily learn the concept as the knowledge level they should specify. This leads us to conclude that there can be a notion of clarity (or vagueness) in specification of the concepts. The experts' estimates were averaged to obtain a composite estimate of knowledge by each learner on each concept. Learners' own estimates were compared against the experts' composite estimate using a paired t-test. Questions 8 and 9 are excluded because students had not learned about the concepts at the time they were interviewed. The t-test result shows a positive correlation between students' estimates and experts' estimates (correlation =0.735), but the inequality of means shows no significant difference (with T=1.491, p<0.15). This indicates that the students' own estimates are not significantly different from the experts' opinions. Note again in Table 2 that this particular student did not agree at all with the experts on item 10. This has more to do with ontological as opposed to cognitive issues. The term ADT was unknown to the student, even though much of the foundational skill on this concept was gleaned by the experts in looking into the portfolio.

Table 2: Sample experiment data for one student and one expert

Concept	Knowledge level Estimated by expert 1	number of artifacts linked by expert 1	Relevance	time (seconds)	Mean knowledge level estimate across experts	StDev knowledge level estimate across experts	Clarity (average of 1 to 5 scale)	Student's estimate of knowledge level
1	3	3	0.59	384	3.6	0.89	4.5	3.5
2	4	2	0.66	573	3.2	0.84	4	2.3
3	3	2	0.42	56	3.4	0.55	3.75	3
4	5	3	0.57	411	3.4	1.34	3.5	4
5	5	1	0.18	140	4	1.00	4.25	3
6	5	1	0.066	120	2.2	1.79	2.5	3
7	5	1	0.13	78	4	1.00	4	3
8	2	2	0.56	199	2	0.00	4.5	N/A
9	2	1	0.56	116	2.2	0.45	3.75	N/A
10	3	0	0	64	2.4	0.89	2.5	1

5.2. Factors that Affect the Accuracy of Evaluation

We attempted to develop a model to characterize the accuracy of an evaluator (experts or possibly student) in judging the learner model after having been initialized from an e-portfolio. It seems that the factors that can affect the accuracy include total time spent on making claims including time to review the potential evidence and attach the artifacts, the total number attached pieces of evidence (attached artifacts), relevance of the evidence selected, and clarity vs. vagueness of the concept to be estimated and justified. The time factor was excluded in this evaluation because the time is considered as a threshold to measure if sufficient attention is spent on making the

claims. All expert evaluators completed the tasks carefully and thoroughly spending adequate time. Further, data collected from web-based and other interactive learning systems, such as detailed logs of page visits, time spent on each page and links selected, give weak evidence that the user read the material or let alone learnt it. [8].

For each claim about one concept, the minimum number of evidence artifacts that the evaluators can choose to attach is zero, and the maximum is the number of artifacts in the course e-portfolio. Usually, only a subset of the e-portfolio artifacts is related to each concept/claim. It seems that the more the number of evidence artifacts attached, the more likely all items in this subset will be covered. However, too many evidence artifacts do not increase confidence in accuracy, and a threshold for the maximum value of the number should be used.

Every time an artifact is selected and attached as evidence, the evaluator considers it related to the current concept (either positively or negatively). However, some evidence can be considered weak evidence because either the content of the artifact is of poor quality or the content is not relevant to the current concept. We tried to determine the relevance of each artifact in the evidence list by asking our expert evaluators to explain the sign of the support (whether it positively or negatively affected the claim) and to provide comments about the evidence attached. The quality of the artifact can be judged based on the frequency of that artifact being selected by any expert evaluator for evidence on any concept, and the relevance to a certain topic can be judged based on that artifact being selected by other/all evaluators for the current concept. We use the product of two normalized frequencies to represent the relevance value of every piece of selected evidence. For each concept, the relevance is the average value of the relevance values assigned to every attached artifact.

A large standard deviation among expert evaluators on “the nested object concept” (concept 6) was found in the result, which reflects that there was some variation in understanding of the intent behind this concept by the expert evaluators (and confirmed in the interview). The clarity of a concept can be defined as the possibility of a concept being interpreted and understood as the expected meaning. Generally the more knowledge about the concept covered in the class and the less abstract the concept it is, it is more easily to be understood in the expected way. Clarity is measured highly depend on the context and rely on the domain concept map and course ontology. The values of clarity of the ten concepts in our accuracy model are defined as the average clarity value from the five invited domain experts (shown in Table 2).

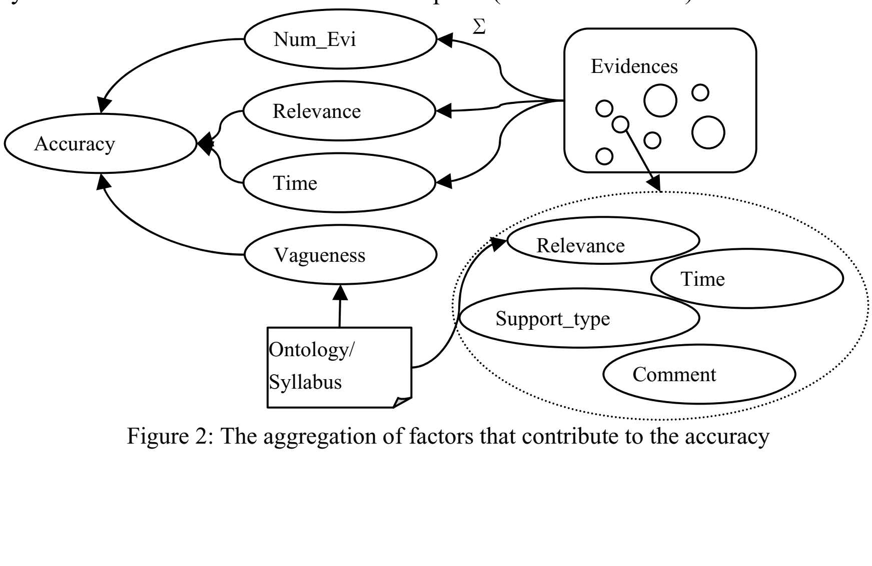


Figure 2: The aggregation of factors that contribute to the accuracy

Figure 2 shows the aggregation of all the factors contributing to the accuracy of the determination of the learner model created through our approach. The various factors affect the accuracy of this determination at different granularity levels. At the higher level, the accuracy of the judgment is determined by the number of attached pieces of evidence (Num_Evi), the relevance of all the evidence, the time spent and the clarity of the concepts affected by domain ontology and curriculum. At the second level, each evidence artifact has four attributes that may contribute in an accumulative or selective way, including relevance of the artifact, time spent on viewing the artifact, support type (positive or negative), and comment.

We propose the following equations to help measure accuracy of knowledge level claims. Average values of claimed knowledge levels are chosen as an accurate basis value because the average results proved reliable in our experiment. The accuracy of each claim (AOC) is defined by equation 1, where x is the rating associated with the claim and μ is the average rating of the five experts. Equation 1 normalizes the values of AOC. NOEc is the total number of evidences attached to a concept. Rc is the relevance of a claim. Cc is the clarity of the concept. Weights of evidence number (Wn), relevance (Wr) and clarity (Wc) can be used as standardized values for judging the accuracy of the process of creating learner models in the same/similar context.

Equation 1: $\mathbf{AOC = 1\text{-}|x\text{-}\mu| \,/\, MAX\,(|\mu\text{-}min|,|\,\mu\text{-}max|)}$

Equation 2: $\mathbf{AOC = NOEc{*}Wn + Rc{*}\,Wr + Cc{*}Wc + Constant}$

To validate this model, we use data for four of the five expert evaluators across all four students and 10 concepts to run a linear regression. The result including weights of Wn, Wr and Wc from the regression is shown in Table 3. It is clear from the result that the number of evidence carries very little weight compared to the relevance and clarity. The result also shows limitations (large standard errors) which are possibly due to our small size of sample data and errors brought by the linear regression. A larger sample dataset and better refined equation will be our future work. To evaluate the usefulness of the weight values, a paired T-test was conducted with the fifth evaluator's judgments to compare the AOC value computed using equation 1 and equation 2. The result shows that the values computed from equation2 using standardized Wn, Wr and Wc values are not significantly different from the AOC values defined by equation1. This means that the accuracy values computed by the model (based on the various factors) are not statistically different from observed accuracy levels. Thus, we can conclude that this model for the accuracy of the evaluation process is promising.

Table 3: Result of the regression and paired T-test

				Wn	Wr	Wc	Constant
Beta Coefficients				-.004	.045	.207	-.027
Std. Error				.009	.09	.027	.09
		Mean	Std. Dev	N	t	Sig.	Correlation
Paired T-test	Pequation1	.7854	.2178	40	-1.331	.191	.657
	Pequation2	.7508	.1467	40			

5.3. Benefit for Reflective Learning

Our experts verified that a significant amount of reflection on prior learning would be achieved if students proceeded through this evidencing process. Asking students to

initialize learner models in this way would lead to a productive learning experience. Yet this analysis raises one additional observation. The artifacts in an e-portfolio should be browseable in human-readable form and this leads to the requirement that learner models, in order to be useful e-portfolio artifacts, be inspectable by users. Research in open learner modelling has shown that inspectable learner models can bring benefits to students and teachers. After completing a session with a learning environment, the learner model could be transferred as a new artifact to the learner's e-portfolio. Attaching inspectable learner models to e-portfolios may provide another means for learner reflection.

6. Conclusion and Future Work

In this paper, we address the interplay between e-portfolios and learner models. We present a system to initialize learner models by making claims about the skill level on various concepts and backing up the claim with evidences drawn from the e-portfolio. An experiment has been conducted aiming at testing how accurately learner models can be created through this approach and if learners can gain benefit in reflective and personalized learning. The results are presented showing its promise in adaptive learning environments.

The next step of the EP-LM system is to add an automated learner information extraction feature that can assist the manual process of making claims about knowledge levels and providing evidence. Information that can be automatically attached includes system information (creation date, ownership, etc.) and personal information (name, address, etc.). Another extension of this research work is to further investigate the possibility of integration or inter-operability with a larger course management system.

References

1. Barrett,H.C. (2001), "Electronic Portfolios; An Encyclopedia to be published by ABC-CLIO", [online]. <http:www.electronicportfolios.com/portfolios/encyclopediaentry.htm>.
2. Grant, P. (2005) Enriching Assessment with ePortfolios, ePortfolio 2005, Cambridge UK. 21-23.
3. Greer, J., & McCalla, G., (1994) Student Models: The Key to Individualized Knowledge-Based Instruction, NATO ASI Series F: Computer and Systems Sciences, vol. 125, Springer Verlag, Berlin Heidelberg.
4. IMS ePortfolio Specification (2005), IMS Global Learning Consortium, Inc. http://www.imsproject.org/ep/index.html, accessed 6/15/2006
5. IMS Learner Information Package Specification (2005), IMS Global Learning Consortium, Inc. http://www.imsproject.org/profiles/index.html, accessed 6/15/2006
6. Pon-Barry, H., Clark, B., Schultz, K., Bratt, E O., Peters, S., & Haley, D. (2005) Contextualizing Reflective Dialogue in a Spoken Conversational Tutor .Educational Technology & Society, 8 (4), 42-51.
7. Guo, Z. & Greer, J., (2005) "Connecting E-portfolios and Learner Models", in proceedings of E-Learn 2005, Vancouver, Canada, October 24-28, 2005, pp 2075-2080.
8. Kay, J., Lum, A. (2005) "Exploiting Readily Available Web Data for Scrutable Student Models", 12th International Conference on Artificial Intelligence in Education, July 2005, Pg 338-345
9. Bull, S., (2004) "Supporting Learning with Open Learner Models", 4th Hellenic Conference with International Participation: Information and Communication Technologies in Education, Athens. Keynote
10. Ebel, Robert L. (1951). Estimation of the reliability of ratings. Psychometrika 16: 407-424.
11. Fleiss, J. L., Cohen, J. (1973). The equivalence of weighted kappa and the intraclass correlation coefficient as measures of reliability. Educational and Psychological Measurement, 33: 613-619.

Linguistics and Language Technology

This page intentionally left blank

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Combining Bayesian Networks and Formal Reasoning for Semantic Classification of Student Utterances

Maxim Makatchev [a,*] and Kurt VanLehn [b]
[a] *The Robotics Institute, Carnegie Mellon University*
[b] *Learning Research and Development Center, University of Pittsburgh*

Abstract. We describe a combination of a statistical and symbolic approaches for automated scoring of student utterances according to their semantic content. The proposed semantic classifier overcomes the limitations of *bag-of-words* methods by mapping natural language sentences into predicate representations and matching them against the automatically generated deductive closure of the domain givens, buggy assumptions and domain rules. With the goal to account for uncertainties in both symbolic representations of natural language sentences and logical relations between domain statements, this work extends the deterministic symbolic approach by augmenting the deductive closure graph structure with conditional probabilities, thus creating a Bayesian network. By deriving the structure of the network formally, instead of estimating it from data, we alleviate the problem of sparseness of training data. We compare the performance of the Bayesian network classifier with the deterministic graph matching-based classifiers and baselines.

Keywords. Dialogue-based intelligent tutoring systems, Bayesian networks, formal methods, semantic classification

1. Introduction

Modern intelligent tutoring systems attempt to explore relatively unconstrained interactions with students, for example via a natural language (NL) dialogue. The rationale behind this is that allowing students to provide unrestricted input to a system would trigger meta-cognitive processes that support learning (i.e. self-explaining) [1] and help expose misconceptions. WHY2-ATLAS tutoring system is designed to elicit NL explanations in the domain of qualitative physics [7]. The system presents a student a qualitative physics problem and asks the student to type an essay with an answer and an explanation. A typical problem and the corresponding essay are shown in Figure 1. After the student submits the first draft of an essay, the system analyzes it for errors and missing statements and starts a dialogue that attempts to remediate misconceptions and elicit missing facts.

Although there are a limited number of classes of possible student beliefs that are of interest to the system (e. g., for the Pumpkin problem, about 20 correct and 4 incorrect

*Correspondence to: Maxim Makatchev, The Robotics Institute, Carnegie Mellon University, 5000 Forbes Ave., Pittsburgh, PA, 15213, USA. Tel.: +1 412 268 3474; Fax: +1 412 624 7904; E-mail: maxim.makatchev@cs.cmu.edu.

Question: Suppose you are running in a straight line at constant speed. You throw a pumpkin straight up. Where will it land? Explain.

Explanation: Once the pumpkin leaves my hand, the horizontal force that I am exerting on it no longer exists, only a vertical force (caused by my throwing it). As it reaches it's maximum height, gravity (exerted vertically downward) will cause the pumpkin to fall. Since no horizontal force acted on the pumpkin from the time it left my hand, it will fall at the same place where it left my hands.

Figure 1. The statement of the problem and a verbatim explanation from a student who received no follow-up discussions on any problems.

beliefs), for each class there are multiple examples of NL utterances that are semantically close enough to be classified as representatives of one of these classes by an expert. Typically the expert will classify a statement belonging to a certain class of student beliefs if either (1) the statement is a re-phrasal of the canonical textual description of the belief class, or (2) the statement is a consequence (or, more rarely, a condition) of an inference rule involving the belief. An example of the first case is the sentence "pumpkin has no horizontal acceleration" as a representative of the belief class "the horizontal acceleration of the pumpkin is zero." An example of the second case is the sentence "The horizontal component of the pumpkin's velocity will remain identical to that of the man's throughout" as a representative of the belief class "The horizontal average velocities of the pumpkin and man are equal": the letter can be derived in one step from the former via a domain rule. The second case occurs due to the coarseness of the classes: the expert would like to credit the student's answer despite the fact that it doesn't match any of the pre-specified semantic classes, based on its semantic proximity to the nearby semantic classes. To summarize, utterances assigned to a particular semantic class by an expert can have different syntactic and semantic features.

While *bag-of-words* methods have seen some successful applications to the problems of semantic text classification [4], their straightforward implementations are known to perform weakly when the training data is sparse, the number of classes is large, or classes do not have clear syntactic boundaries[1] [6]. This suggests using syntactic and semantic parsers and other NLP methods to convert the NL sentences into symbolic representations in a first-order predicate language [7]. However, uncertainty inherent in the various NLP methods that generate those representations [5] means that the same utterances can produce representations of different quality and structure. Figure 2, for example, shows two representations for the same utterance, produced by different NLP methods.

The uncertainty in parsing and in other NLP components adds to the syntactic and semantic variability among representatives of a semantic class. Encoding these sources of variabilities explicitly appears to be infeasible, especially since the properties of the NLP components may be difficult to describe, and the reasoning behind an expert assigning a semantic class label to an input may be hard to elicit. A method to combine different sources of uncertainty in an NLP pipeline using a Bayesian network has been proposed in [3]. Our objective is to adapt this idea to the scenario when semantic classes themselves, as well as relationships between them are uncertain. In particular, we would

[1] Syntactic features alone become insufficient for classification when classes depend on the semantic structure of the domain (as described in the previous paragraph), including the sensitivity to conditionals and negation.

Representation I:
```
(position pumpkin ...)
(rel-position ...pumpkin at)
(quantity2b ...0 ...)
(acceleration pumpkin ...)
(rel-coordinate ...)
```

Representation II:
```
(acceleration man horizontal ...0 ...)
```

Figure 2. Representations for the sentence “There is no horizontal acceleration in either the pumpkin or in the man, and therefore is inconsequential,” produced by two different methods. For the sake of simplicity we omitted uninstantiated variables.

like to learn the relationships between the various elements of symbolic representations and semantic classes directly from the data.

In this paper we propose to combine the expert’s knowledge about the structure of the domain with the structure’s parameter estimation from the data. In particular, we utilize a Bayesian network with nodes representing the facts and domain rule applications, and directed edges representing either an antecedent relationship between a fact and a rule application, or a consequent relationship between a rule application and a fact. In addition, subsets of nodes are grouped as antecedents to the nodes representing semantic class labels. The design of the classifiers is described in Section 2. The details on our dataset are given in Section 3.

The structure of the Bayesian network is derived as a subset[2] of the deductive closure C via forward chaining on the givens of the physics problem using a problem solver, similarly to the approach taken in [2]. However, unlike [2], we estimate the conditional probabilities from the data. We will compare the Bayesian network classifier with (a) a direct matching classifier that assumes syntactic similarity and NLP consistency in generating symbolic representations; (b) a classifier based on the match of the input with a particular subgraph of the deductive closure C and checking the labels in the neighborhood of this subgraph; (c) a majority baseline and (d) a Bayesian network with untrained parameters. The evaluation results are presented in Section 4. In Section 5 we summarize the results and outline the ways to improve the system.

2. Classifiers

In general, our classification task is: given a symbolic representation of a student’s sentence, infer the probability distribution on a set of student beliefs representing knowledge of certain statements about domain. By the domain statements here, we mean physics principles, instantiated principles (facts) and misconceptions. In this paper we will consider an evaluation of the classifying of facts, which is a part of the measure of *completeness*, as opposed to classifying of misconceptions, which we referred to as *correctness* in [8]. We treat these problems differently: analyzing a utterance for misconceptions is viewed as a diagnosis problem (a student’s utterance can be arbitrary far in the chain of

[2]For the sake of simplicity we will refer to the subset of the deductive closure that is fixed throughout the study as the deductive closure.

reasoning from the application of an erroneous rule or fact), while coverage of a fact or a rule by an utterance is viewed as a problem of semantic proximity of the utterance to the fact or the rule. We will compare the performances of the classifiers described below.

2.1. Direct matching

Each of the semantic classes of interest is equipped with a manually constructed symbolic representation of the "canonical" utterance. The direct matching classifier uses a metric of graph similarity based on largest common subgraph [10] and a manually selected threshold to decide whether the symbolic representation of a student's utterance is similar enough to the representation of the canonical member of the semantic class. The method is fast and does not require running any logical inference procedures. However, obviously it can only account for limited variations in the semantic representations.

2.2. Matching against a deductive closure subset of radius 0

For each of the physics problems we generate off-line a deductive closure of problem givens, likely student's beliefs and domain rules. In the case of the Pumpkin problem covered in this paper, the deductive closure has been built up to the depth 6, has 159 nodes representing facts, and contains the facts corresponding to the 16 semantic classes of interest. In fact, out of the total of 20 semantic classes relevant to the solution of the problem we have limited our investigation to the 16 classes that we were able to cover by the deductive closure created with a reasonable knowledge engineering effort. This effort included manual encoding of 46 problem-specific givens and 18 domain rules that can be shared across a range of physics problems.

The nodes of the deductive closure are then automatically labeled (off-line) with the semantic class labels via a graph matching algorithm used in the direct matching classifier described in Section 2.1. During the run-time, the symbolic representation of a student's utterance is matched against the deductive closure via the graph matching algorithm and the matching nodes of the closure are then checked for sufficient coverage of any of the semantic classes by counting their semantic labels [8].

2.3. Matching against a deductive closure subset of radius 1

The radius here refers to the size of the neighborhood (in terms of the edge distance) of the subset of the nodes of the deductive closure that match the symbolic representation of the input utterance. In the previous case, we considered only those nodes that matched the utterance. Here, we consider the set of nodes that are reachable within one edge of the nodes matching the input utterance [8]. Thus the neighborhood of radius 1 contains the neighborhood of radius 0. It is natural to expect that this classifier will over-generate semantic labels, improving recall, but likely decreasing precision.

2.4. Bayesian network, untrained

Our intention is to use the structure of the deductive closure graph to construct a Bayesian network classifier. We augment this graph with additional nodes corresponding to the semantic class instances and semantic class labels. Thus, the resultant Bayesian network graph in our proposed method consists of following types of nodes:

- 159 nodes corresponding to the domain facts in the original deductive closure. These nodes are observed in each data entry: a subset of them that is matched to the symbolic representation of the student utterance is considered to be present in the input utterance, the rest of these nodes are considered not present in the utterance;
- 45 nodes corresponding to domain rule applications in the original deductive closure. These nodes are unobserved. Parents of such node are the nodes corresponding to the antecedent facts of the rule application, and children of such node are the nodes corresponding to the facts generated by the rule application (consequences);
- 16 nodes representing the class label variables. They are childless and their parents are nodes corresponding to the instances of the semantic class among the subsets of fact nodes. These nodes are observed for every data entry of the training set according to the human-generated semantic labels of the utterance.
- 62 unobserved nodes corresponding to the instances of the semantic class in the deductive closure.

Each of 282 nodes in the network is boolean valued. We use informative priors for conditional probabilities: boolean OR for the class label nodes, and boolean OR with probability $p = 0.1$ of reversing its values for the other nodes. In this baseline classifier we don't train the parameters, using just the default conditional probabilities.

2.5. Bayesian network, trained

This classifier consists of the Bayesian network built in the same way and as the classifier above, but this time the network parameters are estimated via Expectation-Maximization (EM) with the same informative priors on conditional probabilities as above, using 90% of the dataset at each step of the 10-fold cross-validation.

3. Dataset

The data set consists of 293 labeled NL utterances collected during a Spring and Summer of 2005 study with student participants. The features are:

- a mapping to the values of the observable nodes of the Bayesian network (deductive closure),
- a human-generated score (an integer between 1 and 7) indicating the quality of the symbolic representation,
- zero or more of human-generated class labels from the set of 16 semantic class labels and their respective binary confidence values (high/low).

From the histogram of the semantic labels shown in Figure 3, it is clear that the data is skewed towards the empty label, with 35.49% of the examples not labeled as corresponding to any of the 16 semantic classes. Thus we expect that the majority baseline classifier that always predicts the empty label would perform quite well. However since it is particularly important to give a credit to a student's contribution when there is one, a slight raise of the performance above the empty-label majority baseline can mean a

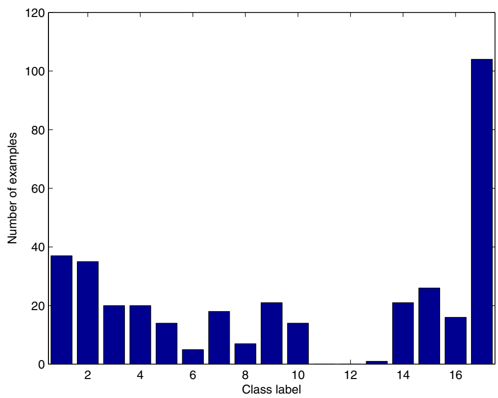


Figure 3. Histogram of the 16 semantic labels and the "empty" label (the rightmost column) in the dataset.

significant change in the application's responsiveness to the student's non-empty contributions.

Due to the relatively small amount of labeled data, for the current experiment we decided to discard the human-generated confidence values of class labels and quality scores of symbolic representation to reduce the dimensionality of the data. However, we account for the degree of confidence in matching the symbolic representations of utterances with the nodes of the deductive closure by generating two instances of the dataset that differ only in the threshold for the graph matching algorithm [10] that decides what nodes of the deductive closure graph correspond to the graph of symbolic representation of the utterance. Namely, for the dataset *data07* the predicate representation of NL utterances is mapped to the nodes of the deductive closure with the more permissive threshold of 0.7 (less overlap of the structure and labels of the two graphs is required), while for the dataset *data09* the threshold is set to the more restrictive value of 0.9 (more overlap of the structure and labels of the two graphs is required). The data with lower confidence are, somewhat counter-intuitively, harder to obtain due to increased search space of the graph matcher. The more permissive similarity matching results in more matched nodes of the deductive closure and therefore of the Bayesian network, potentially making the dataset more informative for training and prediction. This is one of the hypothesis that we test in our experiment in Section 4.

4. Evaluation

The evaluation consists of 10-fold cross-validation on *data07* and *data09* datasets (Tables 1 and 2). The performance measures are average recall and average precision values for each of the entries. The following classifiers have been compared:

- *direct*: Deterministic matching directly to the class representations (no deductive closure structure).
- *radius0*: Deterministic matching to the deductive closure and then checking the labels of the matched closure nodes (uses deductive closure structure).

- *radius1*: Deterministic matching to the deductive closure and then checking the labels of the closure nodes within inference distance 1 from the matched closure nodes (uses deductive closure structure).
- *BNun*: Probabilistic inference using untrained Bayesian Network with informative priors (uses deductive closure structure).
- *BN*: Probabilistic inference using EM parameter estimation on a Bayesian Network with informative priors (uses deductive closure structure).
- *base*: Baseline: a single class label that is most popular in the training set.

Classifier	Recall	Precision	F-measure
direct	0.4845	0.4534	0.4684
radius0	0.5034	0.4543	0.4776
radius1	0.5632	0.4120	0.4759
BNun	0.2517	0.0860	0.1282
BN	0.4948	0.5000	0.4974
Base	0.3897	0.3897	0.3897

Table 1. Performance of 6 classifiers on *data07*.

Classifier	Recall	Precision	F-measure
direct	0.5138	0.5103	0.5120
radius0	0.4690	0.4690	0.4690
radius1	0.4713	0.3957	0.4297
BNun	0.2069	0.0787	0.1140
BN	0.4701	0.4707	0.4704
Base	0.3931	0.3931	0.3931

Table 2. Performance of 6 classifiers on *data09*.

The first observation is that the higher confidence dataset data09 did not result in better performance of the Bayesian network classifier. We attribute this to the fact that high confidence data contained very sparse observations that were insufficient to predict the class label and to train the parameters of the network.

Second, the deterministic methods that take advantage of the deductive closure outperform the deterministic direct matching that does not use the deductive closure on both the recall and precision (*radius0*), or just the recall while sacrificing the precision (*radius1*) (Table 1). Moreover the method that uses a neighborhood of the matching subset of the deductive closure, *radius1*, has better recall (and a worse precision) than the method that uses a just the matching subset of the deductive closure, i. e. *radius0*.

Third, the structure alone is insufficient to improve the precision, since the Bayesian network that doesn't learn parameters (using the default values), *BNun*, performs poorly (worse than the majority baseline).

Lastly, the trained Bayesian network *BN* seems to be best overall, according to the F-measure, at least when the symbolic representation was generated with a more permissive threshold, as in *data07*. However the improvement is modest, and more investigation is required to determine the behavior of the classifier on larger datasets.

5. Conclusion

From a pragmatic stand point, this work has shown that each increment in technology has increased semantic classification accuracy. According to the F-measures, taking an advantage of the graph of semantic relationships (deductive closure) improved the the performance of the deterministic classifiers from 0.4684 (*direct*) to 0.4776 (*radius0*). A Bayesian network that has been built on top of the deductive closure with its parameters estimated via EM-based training outperformed the deterministic methods (with the score 0.4974) in the case when the mapping of the data onto the network is more permissive. Incidentally, this disproved our hypothesis that the training data with higher confidence

in the labels (i. e. less permissive mapping onto the network) must necessarily result in better performance. Finally, we demonstrated a feasibility of deriving of the Bayesian network structure via deterministic formal methods, when the amount of training data is insufficient for structure learning from the data. Although these results are encouraging, they also indicate just how hard this particular classification problem is.

An interesting direction for future work would be to extend the Bayesian network with the nodes representing uncertainty in observations, namely in semantic labels and in symbolic representation. Eventually, we would like to incorporate the Bayesian network in a framework that would guide the tutorial interaction, for example by generating a tutorial action to maximize an information gain or a certain utility function, as in [9].

Acknowledgements

This research has been supported under NSF grant 0325054 and ONR grant N00014-00-1-0600. The authors would like to thank all members of the Natural Language Tutoring group, in particular Pamela Jordan, Brian ‘Moses’ Hall, and Umarani Pappuswamy. The evaluation was done using Kevin Murphy’s Bayes Net Toolbox for Matlab.

References

[1] Michelene T. H. Chi, Nicholas de Leeuw, Mei-Hung Chiu, and Christian LaVancher. Eliciting self-explanations improves understanding. *Cognitive Science*, 18:439–477, 1994.

[2] C. Conati, A. Gertner, and K. VanLehn. Using bayesian networks to manage uncertainty in student modeling. *Journal of User Modeling and User-Adapted Interaction*, 12:371–417, 2002.

[3] Jenny Rose Finkel, Christopher D. Manning, and Andrew Y. Ng. Solving the problem of cascading errors: Approximate bayesian inference for linguistic annotation pipelines. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 618–626, 2006.

[4] Arthur C. Graesser, Peter Wiemer-Hastings, Katja Wiemer-Hastings, Derek Harter, Natalie Person, and the TRG. Using latent semantic analysis to evaluate the contributions of students in autotutor. *Interactive Learning Environments*, 8:129–148, 2000.

[5] Pamela W. Jordan, Maxim Makatchev, and Kurt VanLehn. Combining competing language understanding approaches in an intelligent tutoring system. In *Proceedings of Intelligent Tutoring Systems Conference*, volume 3220 of *LNCS*, pages 346–357, Maceió, Alagoas, Brazil, 2004. Springer.

[6] Claudia Leacock and Martin Chodorow. C-rater: Automated scoring of short-answer questions. *Computers and the Humanities*, 37(4):389–405, 2003.

[7] Maxim Makatchev, Pamela W. Jordan, and Kurt VanLehn. Abductive theorem proving for analyzing student explanations to guide feedback in intelligent tutoring systems. *Journal of Automated Reasoning*, 32:187–226, 2004.

[8] Maxim Makatchev and Kurt VanLehn. Analyzing completeness and correctness of utterances using an ATMS. In *Proceedings of Int. Conference on Artificial Intelligence in Education, AIED2005*. IOS Press, July 2005.

[9] R. Charles Murray, Kurt VanLehn, and Jack Mostow. Looking ahead to select tutorial actions: A decision-theoretic approach. *J. of Artificial Intelligence in Education*, 14:235–278, 2004.

[10] Kim Shearer, Horst Bunke, and Svetha Venkatesh. Video indexing and similarity retrieval by largest common subgraph detection using decision trees. *Pattern Recognition*, 34(5):1075–1091, 2001.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

A Spoken Translation Game for Second Language Learning [1]

Chao WANG [2] and Stephanie SENEFF
MIT Computer Science and Artificial Intelligence Laboratory

Abstract. In this paper, we describe a Web-based spoken translation game aimed at providing language learners with an easily accessible and fun environment to practice speaking the foreign language. Our prototype centers on the task of translating flight domain sentences from English to Chinese. The system presents English sentences as stimuli to elicit Chinese utterances from a user. It tracks a user's performance and rewards them with advancements in difficulty level. A user study was conducted involving 12 learners of Mandarin Chinese. Each participant played the game and answered survey questions. The system achieved 9.7% word error rate on 2834 non-native utterances collected in the user study. All subjects thought the system was helpful at improving their Chinese, and most of them would play it again and recommend it to their Chinese-learning friends.

Keywords. Computer aided language learning, human language technology, machine translation, educational games, human computer interaction

1. Introduction

How to gain and retain skills in a foreign language is a problem facing many language learners. It is generally agreed that living in a country where the new language is spoken is the best way to become fluent [1]. Unfortunately, a total immersion experience is hard to attain, and most students still learn a foreign language in a classroom setting. [2] provides an excellent summary of conditions and pedagogical recommendations for success in a new language, and eloquently argues for the use of computers in foreign language teaching to overcome limitations of the traditional classroom setting.

Speech and language technologies have great potential in Computer Aided Language Learning (CALL) applications. Ideally, a voice-interactive system can role play a language teacher and a conversational partner, to provide the learner with endless opportunities for practice and feedback. In reality, voice-interactive CALL applications are severely constrained by technology limitations [3]: recognition and understanding of non-native speech and robust dialogue modeling remain as unsolved research challenges. A relatively successful application of speech processing technology is in the area of pronunciation training [4,5,6]. In this case, a learner repeats words or phrases prompted by the computer, and receives feedback on the quality of their phonetic pronunciation and

[1]This research is supported by ITRI in Taiwan and by the Cambridge MIT Institute.
[2]Corresponding Author: Chao Wang, 32 Vassar Street, Room 362, Cambridge, Massachusetts 02139, USA. Tel.: +1 617 253 7772; Fax: +1 617 258 8642; E-mail: wangc@csail.mit.edu.

intonation. In contrast, dialogue systems allow language learners more freedom in constructing their own sentences to practice conversational skills. While a number of dialogue systems have been developed (or adapted) for language learning purposes [7,8], the performances are typically too brittle to be widely used by end-users.

This paper explores the use of speech translation technology to develop a novel voice-interactive CALL application [3]. We describe a Web-based computer game, which aims to provide an inviting environment for language learners to practice speaking the new language. By using Web technology [9], the game is easily accessible to anyone with a computer equipped with a microphone and Internet access. Furthermore, no specialty software is required besides a Web browser and the Java run time environment.

Our game prototype is centered around the task of translating spoken phrases and sentences between English and Chinese in the flight information domain. We envision that the translation exercise would serve as a preparation phase for later dialogue interaction with a conversational system for booking flights [10]. The game uses English sentences as stimuli to elicit Chinese speech from the learner. The system keeps track of how many turns a learner takes to complete all the sentences in a game session, and rewards good performance by advancing the learner towards higher difficulty levels. Through the use of speech recognition, language understanding and language generation technologies, the system is able to provide immediate feedback to the learner by paraphrasing his or her utterances in both languages and judging if the perceived translation is correct [11]. The learner can also get help from the system, by asking for a Chinese translation of an English phrase or sentence. The game system utilizes an interlingua-based bidirectional translation capability, described in detail in [12,13].

In the following sections, we will first describe the game system, including its basic features and the underlying technology components. Next, we will report findings of a user study involving 12 participants, who interacted with the system and filled out a survey. We conclude with future plans to extend our work.

2. System Overview

When a student accesses the game's URL, a login page is first presented, where they can enter a user name (for tracking performance), as well as optionally specify the difficulty level and the number of sentences to work on for each game session. Subsequently, the main page (shown in Figure 1) presents a randomly generated task list, and the system prompts the user to translate each sentence in turn. The system paraphrases each user utterance in both languages to implicitly inform the user of the system's internal understanding, and judges whether the student has succeeded in the task. If the user correctly translates the sentence, the system congratulates him/her and advances to the next sentence. Otherwise, the student can click the meta command buttons (e.g., "help me" or "give up") or simply try again. Once the student completes the sentence list, the system summarizes their performance and offers the option to play another round, possibly at a different level of difficulty (up or down). Figure 2 shows an example interaction between the user and the system.

[3] We are not aware of other CALL applications based on a similar paradigm.

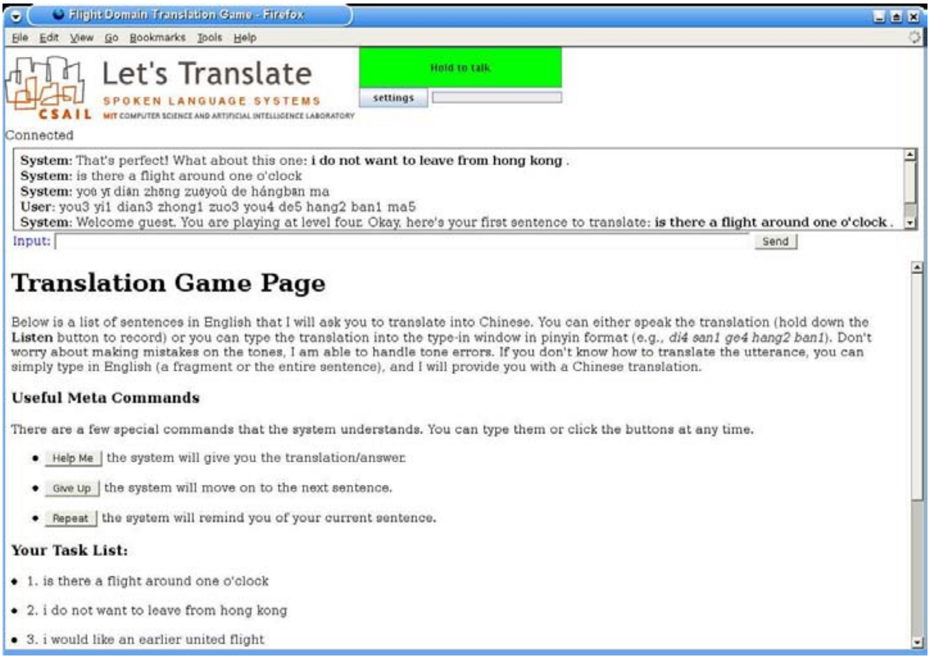


Figure 1. Web interface for the translation game.

System:	(*greet and prompt user*) Welcome! You are playing at level four. Okay, here's your first sentence to translate: **The return date is Tuesday October 25**.
User:	(*translate with error*) fan3 hui2 ri4 qi1 shi4 si4 yue4 er4 shi2 wu3 hao4 xing1 qi1 er4.
System:	(*paraphrase in Chinese*) fan3 hui2 ri4 qi1 si4 yue4 er4 shi2 wu3 hao4 xing1 qi1 er4.
System:	(*paraphrase in English*) The return date is Tuesday April 25.
User:	(*repeat date*) shi2 yue4 er4 shi2 wu3 hao4.
System:	(*paraphrase in Chinese*) shi2 yue4 er4 shi2 wu3 hao4.
System:	(*paraphrase in English*) October 25.
System:	(*respond to user*) 'Great! You successfully translated the sentence in separate fragments. Let's move on to the next sentence: **I need to fly to Hong Kong the day after tomorrow**.
User:	(*click on the help button*)
System:	(*repeat English prompt*) I need to fly to Hong Kong the day after tomorrow.
System:	(*offer example translation*) wo3 xu1 yao4 hou4 tian1 fei1 xiang1 gang3.
User:	(*imitate the translation*) wo3 xu1 yao4 hou4 tian1 fei1 xiang1 gang3.
...	...
System:	(*respond to user*) You translated nine out of ten sentences. You took on average 1.6 turns per sentence. You have advanced to level five. Would you like to play another round?

Figure 2. Example interaction between a user and the system.

2.1. User Utterance Processing

A user's utterance is first processed by the speech recognizer, as shown in Figure 3. The recognizer is based on the SUMMIT landmark-based speech recognition system [14], using Chinese acoustic models trained on native speakers' data [15,16]. Mandarin tone features are ignored in the acoustic models; however, the language model implicitly captures some tone constraints. This is perhaps more appropriate than modeling tone explicitly

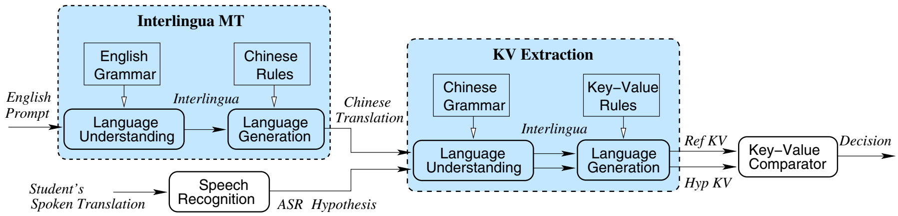


Figure 3. Flow chart of the user utterance verification process.

in the acoustic model, considering that non-native speakers typically have trouble pronouncing tones. The language model was initially trained on Chinese translations of English sentences used in the game. While this achieves good performance when the learners use sentence patterns expected by the system, the performance degrades significantly when they deviate from the expected patterns. We devised a semi-automatic procedure to improve the language model data. The Chinese sentences were converted into templates by using the parser [17] to replace selected words and phrases with variables. This resulted in a much more compact representation of the sentence patterns. We then manually examined the Chinese templates, augmenting the patterns with appropriate alternatives. This process can be iterated as real user data are collected.

After a user's Mandarin utterance is recognized, it is evaluated to determine whether it conveys the same meaning as the English stimulus. This is done via a series of steps as illustrated in Figure 3. The English prompt is first translated into Chinese, and this translation is then compared with the user's translation via an encoding of the meaning in terms of [key: value] (KV) pairs, for ease of scoring. Mismatches are tabulated into substitutions, deletions, and insertions. A perfect match is achieved if the KV pairs of the user sentence and reference are identical. However, partial matches are common, especially when substitution errors on dates and times occur as a result of misrecognition. In such situations, it is natural for the user to just repeat the "incorrect" piece, especially when the sentence is long. Hence, a partial match mode was added to the comparison algorithm, to allow the student to complete the translation in multiple turns. Further details for the assessment algorithm and evaluation results are described in [11].

When the recognizer makes errors, it could result in the system rejecting a perfectly fine user sentence or accepting a wrong answer. Although both types of errors are unfavorable, the first type happens more often and is potentially more frustrating to the user. A well-formed user sentence could even fail in repeated attempts, because of inadequacies in recognition and/or understanding. We provided two mechanisms to work around this condition: the system will move on to the next stimulus sentence if the user is stuck on a sentence for more than a certain number of turns, and the student can also use a meta command to "give up" on the intractable stimulus.

2.2. Game Sentence Inventory

The game system uses a total of over 1000 templates of English sentences in the flight domain, from which it generates the "task list" for each game session. The templates were organized by their length, from short to long, generally reflecting increased difficulty levels. Some manual effort has been devoted to adjusting the order, moving short but linguistically challenging constructs to higher difficulty levels. We harvested these templates semi-automatically from a corpus of real user utterances, obtained from pre-

vious data collection efforts [10]. The templates can also be manually generated if such a corpus is not available. Since not all real user utterances produce good templates, we filter them using the following constraints:

1. The sentence can be fully parsed by the English grammar
2. The Chinese translation automatically generated by the system can be fully parsed by the Chinese grammar
3. The meanings of the sentence pair, encoded as [key: value] pairs, are equivalent.

These constraints aim to ensure that the "correct answer" provided by the system can indeed be accepted by the system. Since the generated Chinese sentences are used in training the recognizer's language model, an utterance is likely to be correctly recognized and accepted whenever the user repeats a provided translation. We verified through our user study that this precaution led to improved usability of the system.

2.3. Game Level Tracking

A user-specific "start index," which controls the difficulty level of the translation task, is retained and updated from session to session. This can be viewed as a simple skill meter model [18], reflecting the student's progress in accomplishing the game task. The index defines a sliding window in the template inventory, which specifies the range of templates from which to randomly generate the task list for a game session. At the end of each session, the system decides on a new start index based on the user's performance in the session, which is represented by how many turns the user takes on average to translate each sentence (asking for help counts as a single turn). In the ideal case, a user can achieve a score of 1.0, i.e, succeeding on the first attempt for each sentence. A heuristic formula is used to update the start index as follows:

$$new\_index = old\_index + (3.0 - score) \times step\_size \tag{1}$$

In our implementation, the $step\_size$ is equivalent to the size of the sliding window, which is set to be 5% of the total number of templates. Hence, the user will stay at the same index if he/she took on average 3.0 turns per stimulus, while a "perfect" performance will advance the user by 10% of the full template space. The index decreases if the user takes more than 3 turns for each sentence. Whenever the change in the start index advances over a 10% interval, the user is notified of a "level" change (often advancing to higher levels). Many participants in our user study liked this feature and commented that it made the game fun. This observation seems to support the recent emphasis on open learner modeling (OLM): exposing the learner model to the learners could promote an individual's reflection on their evolving knowledge and on the learning process [19,20].

When a user's index remains unchanged, the task list for the next session is still very different from the previous one. This is because the "window" of templates is much larger than the number of sentences per game session, and the templates contain substitutable variables such as cities, dates and times. As a result, the game sessions are not too repetitive, even for slow-advancing users.

3. User Study

We conducted a user study involving 12 people who played the game over the Web and filled out a survey afterwards. Figure 4 displays the survey questions. The participants

Tell us about yourself:

1. Have you lived in a Chinese-speaking environment?
2. Is Chinese spoken at home? If so, which dialect?
3. How many years have you studied Mandarin formally? Informally?
4. How do you describe your Chinese proficiency?
5. Have you used other computer programs for learning Mandarin?

How do you like the translation game?

6. Was the game too easy or too difficult? In what ways?
7. Was it fun to play? Why or why not?
8. Did this game help you improve your Mandarin? If so, in what ways?
9. Would you play this game again? Why or why not?
10. Would you play a similar game which covers a different topic? If so, what topics would you like?
11. Would you recommend the system to a friend studying Mandarin? Why or why not?

How can we do better?

12. Would you prefer a more basic game which mainly focuses on vocabulary?
13. Would you prefer a more interactive game, in which the computer is a conversational partner?
14. Please tell us any suggestions which would make the system more enjoyable or effective for you.

Figure 4. Survey questions.

Subject ID	Num Utts	Max Level	WER (%)	Q6	Q7	Q8	Q9	Q10	Q11
A	859	10	9.5	E	+	+	+	+	+
B*	767	10	7.2	R	+	+	+	+	+
C*	228	10	9.8	R	+	+	+	+	+
D	174	10	12.8	E	−	+	o	+	o
E	151	4	10.0	E	+	+	+	+	+
F*	132	5	9.9	D	+	+	+	+	+
G	117	3	15.7	R	−	+	+	+	+
H	109	5	8.1	R	+	+	o	+	+
I	106	1	38.5	D	+	+	+	+	+
J*	73	7	10.3	E	+	+	+	+	+
K	71	3	17.2	D	−	+	−	−	o
L	47	1	15.1	D	o	+	+	+	+

Table 1. Statistics and survey results by user. The participants are ordered by the number of utterances they spoke with the system. The asterisk in the subject ID marks heritage speakers. See text for details.

have varied Chinese background, as indicated by their answers to survey questions 1-5. Four were considered as "heritage" speakers for whom Chinese (including dialects such as Cantonese and Shanghainese) was spoken at home. These speakers are marked by an asterisk in Table 1. Some non-heritage speakers had several years of formal and informal exposure in the past, including living in a Chinese-speaking environment for a month to a year. However, a few stated that they have not been able to keep up with the language in recent years. The most beginner participant (Subject I in Table 1) had just six months of informal exposure. About one third of the participants have used other CALL tools for Chinese, including a flash-card type of software, trial version of Rosetta Stone, and a couple of on-line Chinese dictionary Web sites.

Table 1 summarizes, for each participant, the number of Chinese utterances spoken in total, the maximum difficulty level reached, the speech recognition word error rate (WER), and the answers to survey questions 6 through 11. The WER is the number of errors made by the speech recognizer as a percentage of the number of actual words spo-

ken by a user. The participants are ordered by the number of utterances they spoke with the system. Subjects A and B are students in our group, who tested different versions of the system extensively. The heritage speakers generally have lower WERs compared to other speakers. Subject I has the highest WER, almost four times the average, which correlates with his weak prior exposure. An examination of the system log and the recorded waveforms suggests that this subject mostly imitated the system's prompted answers.

Many users provided detailed comments to the survey questions in addition to yes-no answers. We use "+" for affirmative answers and "-" for negative answers. Some users also expressed mixed opinions (e.g., "yes, if the system improves"), which were mapped to "o" in the Table. The answers to question 6 (whether the game was too easy or difficult) were represented by "D" (too difficult), "E" (too easy), and "R" (about right).

Most participants gave very positive feedback regarding the system. In particular, all users said the system was helpful (Q8 in survey). Among answers to how the system helped them improve their Chinese, the participants listed "refreshing memory on some grammar details," "learning new words and phrases," "expanding alternate phrasing," "forcing me to correct some syllable pronunciations," "learning tones," etc. Eleven out of 12 participants would play the game again, especially if the topics are expanded (Q9 and Q10). Ten of the participants would recommend the system to friends learning Chinese (Q11), the other two would after the system performance is improved.

The participants were divided on whether the game was too easy or too difficult. Four of them thought the game was too easy because "the system advances the user to higher levels too quickly," "the domain is very restricted," and "it is too easy to get help." The game was judged too difficult by Subjects I, K, and L, who also have higher WERs compared to others. A heritage speaker (F*) considered the game to be moderately challenging because of specialized vocabularies such as airline and city names. The rest of our subjects consider the difficulty level just right.

Two thirds of the users thought the game was fun to play (Q7). The other four who did not answer "yes" mentioned that persistent recognition errors were frustrating at times. One also felt that the domain was too restricted. We noted that the word error rates (an indicator of the system's performance for the user) for those four users were among the highest (surpassed only by Subject I, who has very limited Chinese exposure). This seems to suggest that the system performance is a critical factor for user enjoyment.

4. Conclusions and Future Work

In this paper we have described a Web-based voice-interactive CALL system intended to help a native speaker of English learn Mandarin through an intuitive and appealing translation game. Our user studies encourage us to believe that this can be an effective strategy for practicing a language.

In future work, we plan to investigate more sophisticated user models which can capture much richer information about the user's performance. Such information can include the particular vocabulary items or sentence constructs that the student has difficulty with (reflected by user asking for help or by increased number of unsuccessful attempts). The task list can reflect such findings to increase the student's practice in problematic areas in future game interactions.

We also plan to expand the translation game to other domains. Our subjects suggested many interesting topics, such as food, transportation, shopping, family, classroom,

etc. We envision a future version where a suite of topics are offered as options. We also plan to enhance the system's capabilities, including adding Chinese characters in the display, offering a preparation page where a learner can study new vocabularies, and adding a review stage to provide feedback on pronunciation and tones.

While our game is based on a translation task, our goal is to provide an engaging environment for the students to practice speaking the new language. We will look into other more natural ways to illicit learner's speech in the target language, for example, by using pictures. We would also like to assess whether the translation game helps students prepare for interactions with a dialogue system in the new language for booking flights.

References

[1] M. Celce-Murcia and J. Goodwin, "Teaching pronunciation," in *Teaching English as a second language*, M. Celce-Murcia, Ed. Boston: Heinle & Heinle, 1991.

[2] M. Eskenazi, "Using a computer in foreign language pronunciation training: What advantages?," *CALICO Journal*, 16(3), pp. 447–469, 1999.

[3] D. Ehsani and E. Knodt, "Speech technology in computer-aided language learnings: Strengths and limitations of a new call paradigm," *Language Learning & Technology*, 2(1), pp. 54–73, 1998.

[4] M. Eskenazi, "Using automatic speech processing for foreign language pronunciation tutoring: Some issues and a prototype," *Language Learning & Technology*, 2(2), pp. 62–76, 1999.

[5] S. M. Witt, *Use of Speech Recognition in Computer-assisted Language Learning*, Ph.D. thesis, Department of Engineering, University of Cambridge, Cambridge, UK, 1999.

[6] D. Hardison, "Generalization of computer-assisted prosody training: quantitative and qualitative findings," *Language Learning & Technology*, 8(1), pp. 34–52, 2004.

[7] S. Seneff, C. Wang, and J. Zhang, "Spoken conversational interaction for language learning," in *Proc. INSTIL/CALL*, 2004.

[8] W. L. Johnson, S. Marsella, and H. Vihjalmsson, "The DARWARS tactical language training system," in *Proc. Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC)*, 2004.

[9] A. Gruenstein, S. Seneff, and C. Wang, "Scalable and portable web-based multimodal dialogue interaction with geographical databases," in *Proc. of InterSpeech*, Pittsburgh, PA, 2006.

[10] S. Seneff and J. Polifroni, "Dialogue management in the MERCURY flight reservation system," in *Proc. ANLP-NAACL, Satellite Workshop*, Seattle, WA, 2000.

[11] C. Wang and S. Seneff, "Automatic assessment of student translations for foreign language tutoring," in *Proc. NAACL-HLT*, Rochester, NY, 2007.

[12] C. Wang and S. Seneff, "High-quality speech translation in the flight domain," in *Proc. of InterSpeech*, Pittsburgh, PA, 2006.

[13] S. Seneff, C. Wang, and J. Lee, "Combining linguistic and statistical methods for bi-directional English Chinese translation in the flight domain," in *Proc. of AMTA*, Boston, MA, 2006.

[14] J. Glass, "A probabilistic framework for segment-based speech recognition," *Computer Speech and Language*, pp. 137–152, 2003.

[15] C. Wang, D. S. Cyphers, X. Mou, J. Polifroni, S. Seneff, J. Yi, and V. Zue, "MUXING: A telephone-access Mandarin conversational system," in *Proc. ICSLP*, Beijing, China, pp. 715–718, 2000.

[16] H. C. Wang, F. Seide, C. Y. Tseng, and L. S. Lee, "MAT2000 – Design, collection, and validation on a Mandarin 2000-speaker telephone speech database," in *Proc. ICSLP*, Beijing, China, 2000.

[17] S. Seneff, "TINA: A natural language system for spoken language applications," *Computational Linguistics*, 18(1), pp. 711–714, 1992.

[18] A. T. Corbett and J. Anderson, "Knowledge tracing: Modelling the acquisition of procedural knowledge," User Modelling and User-Adapted Interaction, 4, pp. 253-278, 1995.

[19] J. Kay, "Learner know thyself: student models to give learner control and responsibility," in *Proc. of International Conference on Computers in Education*, Kuching, Malaysia, 1997.

[20] S. Bull, M. Mangat, A. Mabbott, A. S. Abu Issa and J. Marsh, "Reactions to inspectable learner models: seven year olds to university students," in *Proc. AIED Workshop on Learner Modelling for Reflection, to Support Learner Control, Metacognition and Improved Communication between Teachers and Learners*, Amsterdam, The Netherlands, 2005.

English ABLE

Diego ZAPATA-RIVERA, Waverely VANWINKLE, Valerie SHUTE, Jody S. UNDERWOOD and Malcolm BAUER
Educational Testing Service, Princeton, NJ 08541 USA

Abstract. Assessment-Based Learning Environments (ABLE) make use of assessment information coming from a variety of sources (e.g., formative and summative) to guide instruction. We have developed English ABLE, an assessment based learning environment designed to help English language learners (ELLs) learn about English grammar. Main features of English ABLE include: Item/task reuse (900 enhanced TOEFL® items were recalibrated based on data from all native Spanish speakers who have taken the test), a Bayesian psychometric student model that makes use item statistics, adaptive feedback, adaptive sequencing of tasks, pedagogical agents (Dr. Grammar, Jorge and Carmen), and an indirectly visible student model. This paper presents English ABLE and reports on some preliminary results from a study that involved 149 native Spanish-speaking ELLs.

Introduction

Researchers in the area of second language acquisition have explored the types of errors that language learners make in relation to their native language (L1), target language (L2), and proficiency level [1-5]. Cowan [1], for example, reports on four major causes for errors made by second language learners: interference from native language, learning strategies that are similar to first language, context of the learning situation, and emotional factors connected with pressure of communication. More recently, Schuster & Burckett-Picker [3] identify six strategies for learning a second language in terms of interlanguage (a dynamic variant of the target language that evolves from an existing knowledge base of the native language): direct use of native language, negative transfer, reduction of grammatical redundancy, simplification, positive transfer, and overgeneralization. Bull [4] also emphasizes the importance of understanding transfer errors and utilizing characteristics of the learner's native language when designing intelligent computer-assisted language learning (ICALL) systems. ICALL systems that make use of student models include: Mr. Collins [5], and I-PETER [6].

We have designed an Assessment-Based Learning Environment (English ABLE) that makes use of student test data gathered from native Spanish speakers and expert opinions to create a Bayesian student model that allows for the development of various advanced adaptive technologies. Specifically, English ABLE features adaptive feedback, adaptive sequencing of enhanced TOEFL® tasks, pedagogical agents, and an indirectly visible student model.

This paper is structured as follows: Section 1 presents English ABLE (i.e., task reuse, Bayesian student model, pedagogical agents and the indirectly visible student model). Section 2 reports on a preliminary analysis of data gathered in November, 2006. We conclude with some final remarks and our plans for future work in this area.

1. English ABLE

English ABLE is an Assessment-Based Learning Environment for English grammar. Assessment-based learning environments make use of assessment information to guide instruction. English ABLE demonstrates the reuse of existing high-stakes tasks in lower stakes learning contexts. English ABLE currently draws upon a database of TOEFL® CBT tasks to create new packages of enhanced tasks targeted towards particular component ELL skills. It presents these packages to learners within an adaptive, scaffolded learning environment to help students master aspects of English grammar. In English ABLE, students try to help a virtual student (Carmen or Jorge) learn English by correcting this student's writing from a notebook of facts (sentences – enhanced TOEFL® tasks). Supplemental educational materials about specific grammatical structures are offered by a virtual tutor (Dr. Grammar). Figure 1 shows a screenshot of English ABLE. The student is helping Jorge find grammatical errors on several sentences. However, he/she has not been successful at finding the errors (only one out of four sentences has been answered correctly). Jorge's knowledge levels, which are also the student's knowledge levels (indirectly visible student model, see section 1.4), show a lack of knowledge for agreement. Jorge seems confused and expresses it ("*I don't understand how to make the verb agree with the rest of the sentence.*"). Dr. Grammar offers immediate verification feedback ("*I see you have selected 'created'. However, this part of the sentence is correct.*"), and additional adaptive instructional feedback (i.e., rules, procedures, examples and definitions).

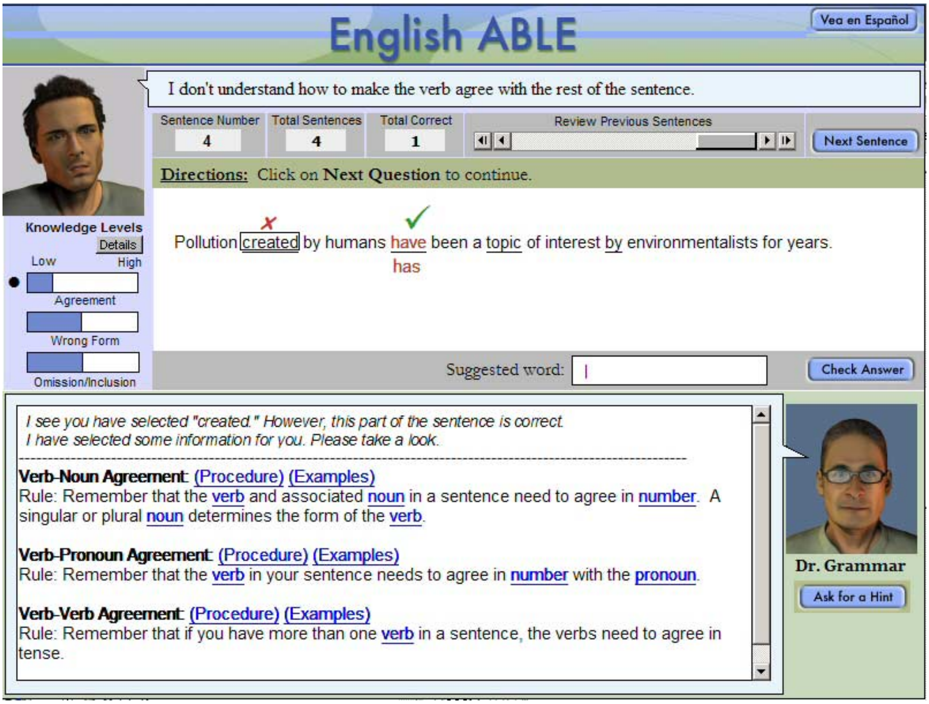


Figure 1. A screenshot of English ABLE

1.1. Task reuse

Existing test items (tasks) are made available to new applications through the use of metadata that is added manually or automatically for a particular purpose. Task

statistics (e.g., difficulty and discrimination parameters), for example, can be used to inform on-line instruction. Additional pieces of metadata include links to relevant learning objects, and a map of concepts (e.g., proficiency map). English ABLE includes approximately 900 enhanced TOEFL® tasks. Packages of enhanced tasks can be used to support a variety of applications including learning environments.

1.2. Bayesian student model

English grammar can be divided into three main categories: use, form and meaning [7, 8, 9]. Working with experts we have elicited an initial Bayesian structure for a student model (see Figure 2). This Bayesian structure can be used to capture and propagate evidence of student knowledge regarding some aspects of English grammar. The current structure of the Bayesian student model deals with English grammar form (although it could be extended to cover use and meaning). Three sentence–level grammatical categories (i.e., wrong form, agreement, and omission and inclusion) have been chosen based upon a difficulty analysis that was performed using student data (i.e. student responses) from only native Spanish speakers. These three sentence-level grammatical categories are further divided into low-level sub-categories (leaf-nodes) according to parts of speech (e.g., agreement has been divided into 3 leaf nodes: noun agreement, verb agreement and pronoun agreement). Leaf-nodes are linked to 2 main knowledge areas (i.e., individual parts of speech: noun, verb and pronoun, and sentence-level grammatical categories) (see Figure 3). Preliminary difficulty analysis plus data from experts were used to generate prior and conditional probabilities for the latent structure. Experts used a qualitative inspired method to produce probability values based on estimates of the strength of the relationship between any two variables in the model [10]. Each task is attached to a single category using existing classification metadata and corresponding IRT (Item Response Theory) parameters (see below).

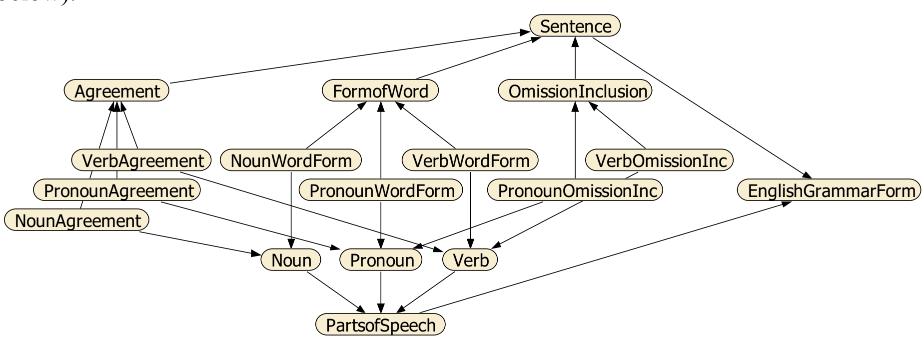


Figure 2. Bayesian student model structure

Tasks were recalibrated based on data from all native Spanish speakers who took the test. Each task is connected to a single leaf-node (proficiency) using the resulting IRT parameters. The IRT-2PL model is described by the following formula:

$$\Pr(task_i = correct \mid proficiency_j) = \frac{1}{1+(e^{-1.701*a(proficiency_j - b)})} \ ,$$

Where b is the difficulty parameter (-3 < = b < = +3, typical values for b), a is the discrimination parameter (-2.80 < = a < = +2.80, typical values for a), and $Proficiency_j$ represents an ability level (continuous proficiency variables were discretized using the following ability values: Advanced = 0.96, IntermediateAdvanced = 0, and

Intermediate = -0.96. These values come from quantiles of a normal distribution [11]). Table 1 shows the resulting conditional probability table of Task 2 (a=1.5, and b= 0.4).

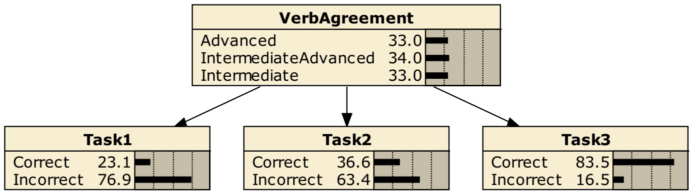


Figure 3. Various tasks attached to VerbAgreement using IRT parameters and organized by difficulty

Table 1: Conditional probability table for Task 2 (a=1.5, b= 0.4)

	Pr(Task2 | VerbAgreement)	
VerbAgreement	**Correct**	**Incorrect**
Advanced	0.807	0.193
IntermediateAdvanced	0.265	0.735
Intermediate	0.030	0.970

As the student makes progress (i.e., answers additional tasks), more tasks/nodes are dynamically added to the model. Observed values per task (i.e., correct or incorrect) provide evidence (as defined by its conditional probability table) to update the student model. Pedagogical agents query the Bayesian student model to provide adaptive feedback, adaptive sequencing of items, and adaptive behavior.

1.3. Pedagogical Agents

Pedagogical agents (e.g., [12, 13, 14]) have been used to facilitate learning by supporting human-like interaction with computer-based systems. Pedagogical agents can act as virtual peers or virtual tutors. Pedagogical agents can model human emotions and use this information to facilitate learning (e.g., [15, 16]). Teachable agents [16], an interesting variant of pedagogical agents, have been used to facilitate student learning. The student's role in these environments is to teach an artificial student how to act in the simulated environment.

Students in English ABLE are asked to help a pedagogical agent (i.e., Carmen and Jorge) find grammar errors. Carmen and Jorge "learn" based on the student's performance. Students can see how much the pedagogical agent knows about a particular concept by looking at the indirectly visible student model (i.e., knowledge levels, see Section 1.4), and by observing Carmen's and Jorge's changes in emotional states and associated utterances.

Carmen and Jorge are able to express basic emotions (see Figure 4), which are triggered by a list of predefined rules. These rules take into account recent performance (e.g., if three sentences attached to the same grammatical structure have been answered correctly in a row, then *emotion* = *happy*) and changes to the student model (e.g., if Pr($proficiency_j$ | *evidence*) ≤ *threshold value*, then *emotion* = *worried*). Utterances for pedagogical agents are dynamically generated and attached to particular emotions. Pedagogical agents in English ABLE are aimed at keeping students motivated.

Dr. Grammar (see Figure 1) provides adaptive instructional feedback (i.e., rules, procedures, examples and definitions) based on the student model. Feedback is not handcrafted at the item level but instead is keyed to grammatical categories, students' native language, and estimated skill level. Amount and type of information presented to

the student varies according to the student's knowledge level (i.e., intermediate students receive more detailed information compared to intermediate-advanced and advanced students).

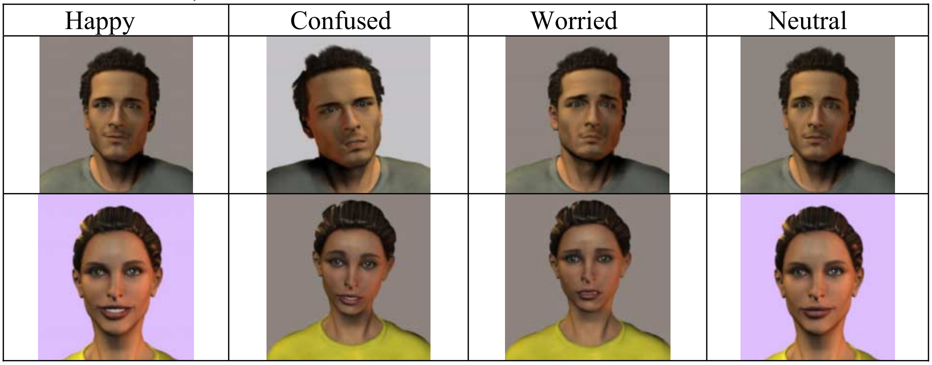


Figure 4. Jorge's and Carmen's emotional states

The difficulty level of the next item (within each grammatical category) is chosen based on short-term past performance and current state of the student model (long-term) following a CAT-like (computer-adaptive testing) approach. Dr. Grammar changes grammatical categories when the marginal probability of $proficiency_i = state_j$ reaches a particular probability threshold value, e.g., 0.80). The next category is selected based on a predefined sequence of categories obtained through preliminary difficulty analysis and mutual information analysis using the Bayesian model.

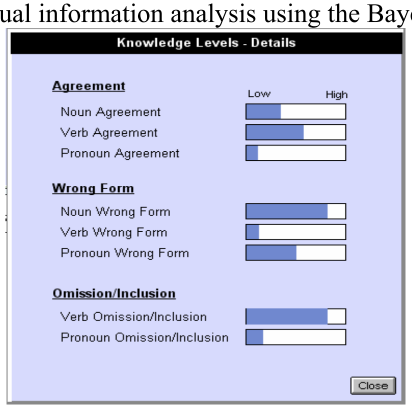


Figure 5. Knowledge levels

1.4. Indirectly visible student model

English ABLE supports indirect inspection of the student model (i.e., knowledge levels in Figures 1 and 5). We consider that exploring one's student model via a pedagogical agent is less intimidating and has the potential to foster student learning without the possible negative effects on self-esteem and motivation, especially for those students who are having a hard time with the system [18]. Previous research on inspecting Bayesian student models through the use of guiding artificial agents showed that agents can facilitate student interaction with the model by helping students navigate and find conflicting nodes. Guided agent interaction was linked to higher levels of student reflection [19].

The length of each bar is determined based on the probability distribution of each node/proficiency using the following formula:

$Length_i = \left[\sum_j^n C_j \Pr(proficiency_i = state_j) \right] / n$, where Cj is a constant numerical value assigned to each state of a node based on its proficiency level (i.e., Intermediate = 0, IntermediateAdvanced = 1, and Advanced = 2) and n is the index of the highest proficiency state (e.g., n = 2, in this case). This produces an Expected A Posteriori (EAP) score that ranges from 0 to 1.

2. The experiment

A study focused on examining usability issues and its effectiveness at helping native Spanish speakers learn English grammar was carried out in November, 2006. The current analysis includes data from 149 participants assigned to 3 conditions (see table 3). Participant institutions include 1 high school, 4 community colleges, 1 college/university, and 2 adult learning centers in four different states (i.e., NJ, PA, NY, and MD). The majority of our sample was comprised of community college students. Participants' ages ranged from 15 to 73 years old with a mean age of 33 and a mode of 21 years of age (all the participants were native Spanish speakers).

2.1. Experimental design

Each session consisted of a 20-minute pretest, a 1-hour intervention (*conditions 1*, *2* or *3*), a 20-minute posttest, and a 10-minute usability survey. Conditions 1, 2 and 3 had the features described in the table below included in the learning tool to different degrees: interface, student model, and pedagogical agents.

Table 2: Experimental design

Features	Condition 1 (control)	Condition 2	Condition 3
Interface	Test preparation environment (text-based)	English ABLE simple	English ABLE enhanced (visible student model, hints and making corrections)
Student Model	None	None	Bayesian student model
Pedagogical Agents	None Verification feedback available. Predefined, pedagogically-sound task sequencing.	Carmen/Jorge (limited interactions based on short-term performance – last 3 sentences) Dr. Grammar (verification feedback only, predefined, pedagogically-sound task sequencing)	Carmen/Jorge (interactions based on student model – short and long-term) Dr. Grammar (verification and adaptive instructional feedback and adaptive task sequencing based on the student model)

2.2. Results

This section presents some preliminary results.

Usability information –Pedagogical Agents

Participants (88% or more across all three conditions) thought that the feedback provided by the system was useful. Regarding Jorge's and Carmen's visible student model (*condition 3*), most participants (88%) stated that they understood the knowledge levels, 86% thought that the knowledge levels were useful, and 86% agreed that the knowledge levels helped them understand what Jorge/Carmen knew. In general, participants thought that they had learned a lot from using the software (81% or more across all three conditions).

Participants in *conditions 2* and *3* agreed with the following statements: (a) "*I liked helping Carmen/Jorge find grammar errors*" (95% and 89%, conditions 2 and 3

respectively), (b) "*Carmen's/Jorge's comments were useful*" (75% and 81%, respectively), (c) "*Helping Carmen/Jorge motivated me to keep going*" (88% and 92%, respectively), (d) "*I have helped Carmen/Jorge a lot by finding the grammar errors*" (75% and 71%, respectively), (e) "*I have learned by helping the Carmen/Jorge with his/her sentences*" (92% and 89%, respectively), (f) "*The feedback provided by Dr. Grammar helped me learn*" (96% and 78%, respectively), and (g) "*I think Carmen and Jorge liked my help*" (83% and 79%, respectively). Because of the exploratory nature of our usability study, we did not do statistical analysis on these data.

Learning effects

We report on the following three research questions:

1) *Condition.* What is the effect of increasing the number of features (*conditions 1*, *2* and *3*) in the English ABLE environment on learning outcome?

2) *ESL level.* What is the relationship between ESL level on learning outcome in English ABLE?

3) *Condition x ESL Level.* Are any of the conditions more or less supportive of learners at different levels of their language proficiency?

To answer these questions, we computed an ANCOVA using *posttest correct* as our dependent variable, and *condition* (*1*, *2*, and *3*) and *ESL level* (Intermediate and Advanced)[1] as our independent variables. To control for differences in (a) incoming knowledge, and (b) number of posttest items actually solved, we used *pretest correct* and *posttest items answered* respectively as covariates in the analysis. The results of the ANCOVA predicting learning outcome were as follows: (a) significant main effect of *condition* ($F_{2,\,129} = 3.42$, $p < 0.05$), (b) significant main effect of *ESL level* ($F_{1,\,129} = 12.18$, $p < 0.01$), and (c) no significant interaction between *condition* and *ESL level*. Estimated means, standard errors, and sample sizes for the posttest-correct data, separated by condition and ESL levels are presented in Table 3.

Table 3: Estimated Marginal Means for Posttest Correct (with Standard Error and N)

Condition	Intermediate	Advanced
1. Test preparation	**13.6** (0.7, 36)	**15.2** (1.3, 12)
2. English ABLE simple	**14.1** (0.8, 26)	**16.3** (0.9, 24)
3. English ABLE enhanced	**14.6** (0.8, 30)	**19.8** (1.4, 9)

Regarding the significant main effect of condition on outcome, Table 3 suggests that more sophisticated versions of the program yield progressively better learning outcomes. To test this assertion, we computed a planned comparison (Least Significant Difference) on these data and results showed that *condition 1* was significantly different from *condition 3* (mean difference = -2.8, $SE = 1.1$, $p < 0.05$), and *condition 2* was marginally different from *condition 3* (mean difference = -2.0, $SE = 1.0$, $p = 0.052$). And while *conditions 1* and *2* were not significantly different from each other, the trend is in the right direction. Finally, the main effect of ESL level indicates that the more advanced students demonstrated greater learning, overall, compared to their intermediate counterparts.

3. Conclusions and future work

Preliminary findings reported in this paper show that English ABLE has potential for facilitating student motivation and learning. We just started exploring the benefits of

[1] The sample size for our self-reported Beginners was very small ($n = 12$), particularly within some of the conditions (e.g., test prep $n = 3$ and English ABLE simple $n = 2$), so we eliminated them from the current analysis.

English ABLE. A longer study would provide more information regarding the adaptive, persistent features of English ABLE enhanced (*condition 3*). Additional reflection tasks can also be included, which would make the profile of pedagogical agents more prominent. Initial results demonstrate that English ABLE has a role to play in helping ELLs learn English grammar. Future work would include adapting the Bayesian model and feedback to various native languages.

References

[1] Cowan, J.R. (1976, July). Toward a psychological theory of interference in second language learning. Paper presented at the Conference on Second Language Learning and Teaching, New York, NY.

[2] Schuster, E. (1986). The role of native grammars in correcting errors in second language learning. *Computational Intelligence*, 2(2), 93-98.

[3] Schuster, E., Burckett-Picker, J. (1996). Interlanguage errors becoming the target language through student modeling. *Fifth International Conference on User Modeling*, Kailua-Kona, Hawaii.

[4] Bull, S. (1995). Handling native and non-native language transfer in CALL: Theory and practice. R. Wakely,

[5] Bull S. & Pain H. (1995), "'Did I say what I think I said, and do you agree with me?': Inspecting and Questioning the Student Model", *Proceedings of World Conference on Artificial Intelligence in Education (AACE)*, Charlottesville, VA, 501-508.

[6] Read, T., Barcena, E., Barros, B., & Verdejo, F. (2002). Adaptive modelling of student diagnosis and material selection for on-line language learning. [On-line version]. *Journal of Intelligent & Fuzzy Systems*, 12, 135-149.

[7] Celce-Murcia, M., & Larsen-Freeman, D. (1999). The Grammar Book (2nd ed.). Heinle & Heinle Publishers.

[8] Greenbraum, S. (1996). The Oxford English Grammar. New York: Oxford press.

[9] Gucker, P.(1966). Essential English Grammar. NJ: Dover, Publications. New York and London: Dover Publications.

[10] Daniel, B., Zapata-Rivera, J.D., McCalla, G. (2003) A Bayesian Computational Model of Social Capital in Virtual Communities. *Proceedings of the First International Conference on Communities and Technologies; C&T 2003*. Kluwer Academic Publishers. ISBN 1-4020-1611-5.

[11] Almond, R., Dibello, L., Jenkins, F., Mislevy, R., Senturk, D., Steinberg, L., and Yan, D. (2001) Models for conditional probability tables in educational assessment. *In T. Jaakkola and T. Richardson, editors, Artificial Intelligence and Statistics*, 137–143.

[12] Chan, T. W. & Baskin, A. B. (1990). Learning Companion Systems. In Frasson, C., and Gauthier, G. (eds.) *Intelligent Tutoring Systems: At the crossroads of AI and Education*. Ablex Pub., 6-33.

[13] Graesser, A. C., Person, N., Harter, D., & TRG (2001). Teaching tactics and dialog in AutoTutor. *International Journal of Artificial Intelligence in Education*, 12, 257-279.

[14] Johnson, W. L.; Rickel, J. W.; and Lester, J. C. (2000). Animated Pedagogical Agents: Face-to-Face Interaction in Interactive Learning Environments. *International Journal of Artificial Intelligence in Education*, 11(1), 47-78.

[15] Picard, R.W. (1997). Affective computing. Cambridge, MA: MIT Press.

[16] Biswas, G., Schwartz, D., Bransford, J., and the Teachable Agent Group at Vanderbilt (TAG-V) (2001). Technology support for complex problem solving: From SAD environments to AI. *In K. D. Forbus and P. J. Feltovich (Eds.), Smart machines in education: The coming revolution in educational technology.* Menlo Park, CA: AAAI/MIT Press. 71-97.

[17] Nkambou, R., Laporte, Y., Yatchou, R., and Gouardères, G. (2003). Embodied emotional agent in intelligent training system. *In Recent Advances in intelligent Paradigms and Applications, J. Kacprzyk, A. Abraham, and L. C. Jain, Eds. Studies In Fuzziness And Soft Computing.* Physica-Verlag Heidelberg, 235-253.

[18] Bull, S., Mangat, M., Mabbott, A., S. Abu Issa, A. & Marsh, J. (2005). Reactions to Inspectable Learner Models: Seven Year Olds to University Students. *AIED'05 Workshop on Learner Modelling for Reflection, to Support Learner Control, Metacognition and Improved Communication between Teachers and Learners*. 23-32

[19] Zapata-Rivera, J.D., & Greer, J. (2004) *Interacting with Bayesian Student Models. International Journal of Artificial Intelligence in Education.* Vol. 14, Nr. 2 127-163.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Using Machine Learning Techniques to Analyze and Support Mediation of Student E-Discussions

Bruce M. MCLAREN[1], Oliver SCHEUER[1], Maarten DE LAAT[2], Rakheli HEVER[3], Reuma DE GROOT[3], and Carolyn P. ROSÉ[4]
[1] *Deutsches Forschungszentrum für Künstliche Intelligenz, Saarbrücken, Germany*
[2] *Exeter University, United Kingdom*
[3] *The Hebrew University of Jerusalem, Israel*
[4] *Carnegie Mellon University, Pittsburgh, USA*
{Bruce.McLaren, Oliver.Scheuer}@dfki.de; M.F.DeLaat@exeter.ac.uk; mt.scopus@gmail.com; msruma@mscc.huji.ac.il; cprose+@cs.cmu.edu

Abstract. Students are starting to use networked visual argumentation tools to discuss, debate, and argue with one another about topics presented by a teacher. However, this development gives rise to an emergent issue for teachers: how do they support students during these e-discussions? The ARGUNAUT system aims to provide the teacher (or moderator) with tools that will facilitate effective moderation of several simultaneous e-discussions. Awareness Indicators, provided as part of a moderator's user interface, help monitor the progress of discussions on several dimensions (e.g., critical reasoning). In this paper we discuss preliminary steps taken in using machine learning techniques to support the Awareness Indicators. Focusing on individual contributions (single objects containing textual content, contributed in the visual workspace by students) and sequences of two linked contributions (two objects, the connection between them, and the students' textual contributions), we have run a series of machine learning experiments in an attempt to train classifiers to recognize important student actions, such as using critical reasoning and raising and answering questions. The initial results presented in this paper are encouraging, but we are only at the beginning of our analysis.

1. Introduction

It is becoming increasingly common for students to use computer-based tools to discuss, debate, and argue with one another about topics presented in a classroom. Such collaborative software tools are designed to allow students to work on separate computers but communicate in synchronous fashion, contributing to an evolving discussion through a shared "workspace." In Israel, England, and the Netherlands, for instance, we are working with and collecting data from over 15 classrooms that are using the tool Digalo (http://dito.ais.fraunhofer.de/digalo/) to engage students in e-discussions. We also have plans to collect data from the use of another collaborative tool, Cool Modes (www.collide.info/software). These tools share a common model in supporting e-discussions: a graphical environment with drag-and-drop widgets that students can use to express their ideas, questions, and arguments in visual fashion.

Of course, simply providing students with computer-based tools, such as Digalo and Cool Modes, will not lead to fruitful discussion and collaboration. Evidence from the computer-supported collaboration literature suggests that fruitful collaboration does not occur spontaneously [1]. One approach that has been used and continues to be investigated by many researchers is the notion of providing a "script" to support collaboration (e.g., [2, 3]). Another approach is to provide a software agent that can coach and/or tutor the collaborating students [4]. A third approach is to include an artificial student in the collaboration whose responsibility it is to provide student-like

contributions and peer coaching [5].

Yet another approach, taken within the ARGUNAUT project discussed in the current paper, is to assist the human moderator, or teacher, of an e-discussion in keeping students on topic, correcting misstatements, and generally guiding the students toward fruitful discussion and collaboration [6]. Such an approach has the advantage of leveraging the knowledge and skills of perhaps the premier resource in supporting learning and collaboration: the teacher. To the extent a teacher can provide personalized attention to individual conversations, and individuals in those conversations, he or she is *tutoring* the students, widely considered to be the most effective way of providing instruction.

However, the cognitive load of moderating multiple simultaneous e-discussions may be too high on teachers. The ARGUNAUT system aims at relieving this by providing the teachers with feedback regarding important elements of each discussion, explicitly focusing their attention on events or situations requiring their intervention.

An approach we are pursuing is to use machine-learning techniques [7] to evaluate past e-discussions and use the results to provide "Awareness Indicators" for teachers and moderators in the context of new e-discussions. Pedagogical experts on our team have annotated components of past discussions as to whether, for instance, students applied critical reasoning, were on topic, or were engaged in raising and answering questions. We then used these annotations to train machine-learning classifiers for the purpose of identifying these meaningful characteristics in new e-discussions. The annotations and subsequent classifications are based on structural, process-oriented, and textual elements of the student contributions.

In this paper, we provide an overview of ARGUNAUT, present and discuss the initial results obtained in applying machine learning to a corpus of Digalo data for the purpose of supporting ARGUNAUT Awareness Indicators, and discuss our next steps in attempting to realize fully the potential of machine learning applied to e-discussions.

2. Overview of the ARGUNAUT Project

An e-discussion in ARGUNAUT proceeds by students responding to a given question (e.g., "What is your opinion about experiments on animals?") and to one another's contributions. Students make contributions to the discussion by dragging and dropping shapes corresponding to discussion moves, such as "question" or "claim," filling text into those shapes to express an idea or argument, and linking shapes to other shapes with directed or undirected links, labeled as "supports" or "opposes." To support the moderator of such discussions, the approach we have taken is to provide a "Moderator's Interface" that contains, among other data, a series of "Awareness Indicators" associated with each e-discussion. The Moderator's Interface allows the teacher to oversee all of the e-discussions currently taking place in the classroom.

Each Awareness Indicator of an e-discussion provides summarized information about an important aspect of the discussion. An indicator may be a relatively easily computed aspect of the e-discussion, such as how many times each student has contributed to the conversation, or a more complex analysis component, such as whether the students have been or are currently engaged in critical reasoning. The human moderator is the ultimate judge of when and how to intervene in an e-discussion; the indicators are intended only to call attention to potentially important discussion characteristics.

The architecture we have designed and partially implemented to support this approach is shown in Figure 1. The end user environment, shown on the left side of

Figure 1, is the Digalo or Cool Modes tool used by groups of collaborating students. All actions taken by the students, such as creating a new shape, link, or textual contribution, are logged. The moderator (teacher) uses the Moderator's Interface, shown in the middle of Figure 1, to monitor on-going e-discussions and intervene when appropriate. The Awareness Indicators are based on either "Shallow Loop" analysis, straightforward calculations, such as counting the contributions made by each student, or "Deep Loop" analysis, application of machine-learning classifiers to the logged data. Using the Annotation Tool, the moderator can annotate maps in real-time to provide additional data to train the Deep Loop classifiers. The Deep Loop, shown on the right side of Figure 1, is an off-line process that takes annotated maps, translates them to a form suitable for machine learning, and generates new classifiers that are subsequently used at run time to classify actions of the students in real time.

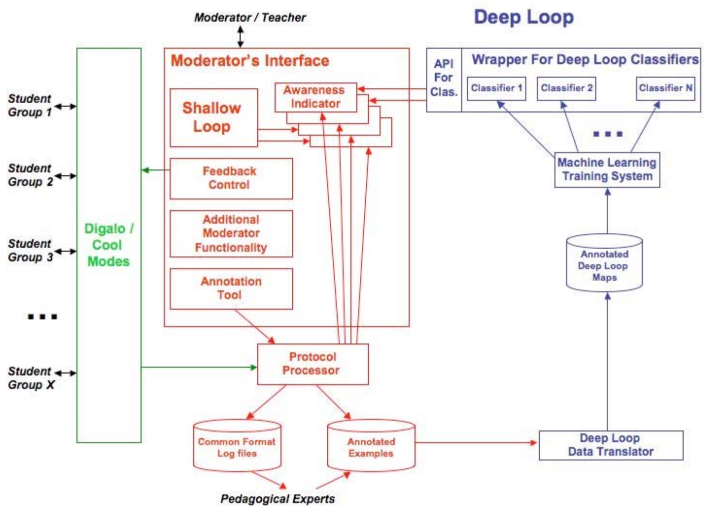


Figure 1: The ARGUNAUT Architecture

3. Annotating Collaborative Data for the Deep Loop

The first step in generating Deep Loop classifiers for the ARGUNAUT system was the manual annotation of existing e-discussion Digalo maps. This corresponds to the process depicted at the bottom of Figure 1, in which human pedagogical experts annotate real maps. The annotated maps are the products of actual Digalo sessions in Israeli junior high school and university classrooms. For the purposes of our initial analyses, the maps were also translated from Hebrew to English.

The experience of our pedagogical experts suggested general criteria for coding, such as participation and responsiveness [8], as well as dialogic analysis [9]. The selection of annotation categories was based on discussion between the pedagogical and technical experts of our team, as well as an iterative analysis of the maps.

We developed specific annotation schemes for two aspects of the Digalo maps: (1) individual contributions (*the shape level*, i.e., single shapes) and (2) connected contributions of one or more students (the *paired-shapes level,* i.e., two contributions (shapes) with a connecting link) in a map. While the shape-level annotations focus primarily on interpretation of the text within a shape, the paired-shapes level involves analysis of structural, process-oriented, and textual aspects of the shapes. A paired-

shape involves the interpretation of two distinct but related pieces of text, the structural relationship between the contributions (a connector), and the order of the contributions. In total, we defined seven annotation variables for the shape level (topic focus, task-management focus, critical reasoning, request for clarification or information, critical evaluation of opinions, summary, and intertextuality) and five annotation variables for the paired-shapes level (question-answer, contribution-counterargument, contribution-supportingargument, contribution followed by question, and qualifier/compromise). Tables 1 and 2 show examples of a few of these variables.

Table 1: Sample of the Shape-Level Coding Scheme

Annotation-variable name	**Explanation / Coding**	**Examples**
Topic focus (TF)	A contribution that focuses on the topic or task. 0. No topic focus content 1. Topic focus content	TF: "Its not nice of human beings to exploit animals for their own needs. I think animals also have rights." Non-TF: "I'm bored."
Critical Reasoning (CR)	An individual contribution that contains critical reasoning or argumentation (i.e., claim + backing). Student provides an explanation or some backing (e.g. evidence) to illustrate a position/opinion. If you can add "because" between two parts of the contribution, it is probably critical reasoning. 0. No critical reasoning 1. Use of critical reasoning	CR: "I am against experiments on animals, because to my opinion it is not fair to use them against their will while they cannot reject." CR: "Here it's not like with humans, as the father disengages from them, and he doesn't see them even in the afternoon, and he doesn't belong to the pack any more"

Table 2: Sample of the Paired-Shapes Coding Scheme

Annotation-variable name	**Explanation / Coding**	**Examples**
Contribution-CounterArgument (CCA)	A contribution in which the 2nd shape opposes the claim/argument raised in the 1st shape and provides reasons or other type of backing for the opposing claim. Typically (but not necessarily) the type of link between the shapes would be "opposition." 0. Not a Contribution-CounterArgument 1. A Contribution-CounterArgument	1st shape text: "Do not separate, the male should be a partner in what happens even after the birth. The offspring is also his and he should take responsibility." 2nd shape text: "But in a situation like this the mother can get pregnant again and so might neglect a group of cubs." Link between shapes is "opposition"
Question-Answer (QA)	A contribution in which the 1st shape is a question, the 2nd shape is an answer to that question. Typically (but not necessarily) the type of link between the shapes would be "other." 0. Not a Question-Answer 1. A Question-Answer	1st shape text: "In the wild does the father separate from the cubs or does he continue to live with them?" 2nd shape: "They all live in a pack" Link between shapes is "other."

At the shape level, a total of 677 shapes in 42 Digalo discussion maps were annotated. Three skilled coders initially applied the coding scheme summarized in Table 1 to a small number of maps. Coding differences were resolved through discussion and decision rules, and the annotation scheme was revised. The three coders then annotated the remainder of the maps, with discussion and decision rules resulting in a "consensus" annotation for every shape of every map. Interrater reliability, using Fleiss' Kappa, was calculated for the second round of annotations, resulting in an acceptable value (or within range of acceptability) for all variables (i.e., close to 0.7).

At the paired-shapes level, 226 paired-shapes in 21 Digalo discussion maps were annotated. Two of our coders applied the coding scheme summarized in Table 2 to five maps, resolved differences, and applied a revised coding scheme to the remainder of the maps. Interrater reliability, using Cohen's Kappa, was calculated, with acceptable values for all variables except qualifier/compromise, which was not used for learning.

4. Use of Machine Learning Techniques in the Deep Loop

Given these manual annotations, our ultimate goal is to train machine-learning classifiers to predict the appearance of these discussion characteristics in the context of new e-discussions and make the resultant classifiers available to the ARGUNAUT Awareness Indicators depicted in Figure 1. Our first step toward this goal was to validate the use of a machine learning approach and to identify the variables that hold the greatest promise of being useful to the Awareness Indicators. It is clearly non-trivial to generate good classifiers, given the complex characteristics of e-discussions, including the shapes chosen by students for contributions, the links created between shapes, and the text provided within a shape. On the other hand, an approach that does reasonably well, with a success rate of, say, 80% to 90% in the most critical situations, may be sufficient for the purposes of moderation.

Our approach draws from and is most closely aligned with that of Donmez, Rosé, Stegmann, Weinberger, & Fischer [10]. Using TagHelper, a software tool that leverages machine-learning techniques to classify text, they ran multi-dimensional analyses over a large corpus of textual argumentation data and achieved a 0.7 Cohen's Kappa (or higher) on 6 of 7 dimensions. We are also doing multi-dimensional analysis, as exemplified by the multiple variables of interest at the shape and paired-shapes levels, and also experimented with TagHelper as part of our analysis. On the other hand, our objective is different. Donmez *et al* focused exclusively on textual contributions, while we are interested in structural and chronological data in addition to the text. Also, while Donmez *et al.*'s goal is to automate corpus analysis for coders, we will use our classifiers to provide feedback for e-discussion moderators.

5. Initial Analyses and Results of Machine Learning

For our initial analyses, we experimented both with TagHelper and YALE, freely available software that supports interactive experimentation with a wide range of machine learning algorithms [11]. We ran some preliminary YALE experiments, using the first 21 annotated maps (318 shapes). We derived a basic set of attributes that expressed characteristics of the text and structure of the shapes, including text length, shape type, number of in-links, number of out-links, and number of undirected links. Using a variety of algorithms, including J48 decision tree and PART, and 10-fold cross validation, the critical reasoning variable yielded classifiers that had by far the best results, including a Kappa value of 0.72 (above the standard acceptable level of 0.7) with most other Kappas at or near acceptable (0.6 to 0.7). All of the other shape-level variables yielded classifiers with lower Kappa values, 0 in many cases. A possible reason for the lower Kappa values is the proportionally unbalanced annotations of either positive or negative labels for these other variables, compared to the relatively balanced distribution of positive (47%) and negative (53%) labels for critical reasoning.

We then focused solely on the critical reasoning variable, since it appeared to hold the best prospects for the machine learning approach (and is also a variable of keen interest to our pedagogical experts). We scaled up our analysis to the fully annotated 42 maps (677 shapes) and experimented with deeper aspects of the language within shapes. In addition to the basic set of attributes described above, we derived two additional attributes that represent word-level contributions: *term_evidence_pos*, a count of the number of words used in the text contribution of a shape that are contained in a "positive" term list, and *term_evidence*, the difference between the number of words used in the text contribution that are contained in the "positive" and the "negative" term lists. The two lists consisted of the top 25 words in the "positive" and "negative" categories identified by a word analysis of the entire corpus of 677 shapes,

using the Odds Ratio (originally defined, but not named, in [12]). We also applied TagHelper to the data, and it generated over 1600 text-focused attributes, representing terms and parts of speech in the corpus.

Table 3: ML Results for the Critical Reasoning Var. running AdaBoost with DecisionStump over 42 Maps

Attributes	Parameters	True Pos. Rate	True Neg. Rate	Acc.	Kappa
Basic set (baseline)	I = 10 (Default)	84%	74%	79%	**0.57**
Basic set	I = 100 (Tuning)	81%	81%	81%	**0.62**
(+ *term_evidence_pos*)	I = 100 (Tuning)	80%	83%	82%	**0.63**
(+ *term_evidence*)	I = 140 (Tuning)	79%	84%	82%	**0.63**
(+ TagHelper attributes)	Chi-squared att. sel.; I = 60 (Tuning)	82%	87%	85%	**0.68**

A summary of the results of these experiments, again run using 10-fold cross validation, are shown in Table 3. AdaBoost with DecisionStump yielded the best results without parameter tuning, so we focused on this algorithm in our experiments. As can be seen, the baseline Kappa achieved with this algorithm, with no parameter tuning run over the basic set of attributes (i.e., the "Default"), was 0.57. Adding the *term_evidence_pos* and *term_evidence* attributes to the basic set of attributes, as well as applying parameter tuning to the number of AdaBoost iterations (I), yielded improvement over the baseline, but the best Kappa (0.68) was obtained using chi-squared attribute selection over the TagHelper and basic attributes. The difference between the baseline and TagHelper Kappa is significant ($p < 0.003$) using a t-test. Thus, our initial shape-level experiments demonstrate that at least for the critical reasoning variable, we can obtain an acceptable Kappa value with the assistance of TagHelper. These experiments also demonstrate the potential value of text-focused attributes combined with structural attributes when learning over our e-discussion maps, as the use of TagHelper attributes yielded the best results.

Finally, we performed a series of YALE experiments to investigate how well we could train classifiers to identify the paired-shapes variables, such as those shown in Table 2 (e.g., question-answer), using the 21 annotated maps (226 paired shapes) as training data. In this analysis we accounted for all three aspects of the maps – text, structure, and process. However, we did not use TagHelper, as it seemed to have a somewhat less logical application to multiple texts associated with single training instances (i.e., the separate texts associated with each of the shapes in a paired shape). We also tried to see if we could re-use the shape-level annotations as attributes in these learning experiments. At the paired-shapes level, we identified and experimented with three sets of mutually exclusive attributes, as well as combinations of those sets:

1. *Basic attributes that do not represent process or structure:* verticesSameUser, combinedTextLength, diffInTextLength, linkType, timeBetweenFirstModOfEachShape
2. *Attributes that represent ordering (i.e. process) and structure of shapes:* link_v[1,2]_sameUser, textLength[1,2], shape[1,2], linkDirection
3. *Attributes that represent ordering and rely on shape-level annotations (using the human annotations):* vertex[1,2]_TopicFocus, vertex[1,2]_TaskManagementFocus, vertex[1,2]_CriticalReasoning

Table 4 shows the results we achieved for the question-answer variable, which had the most balanced proportion of positive (29%) and negative (71%) examples, using the PART algorithm, 10-fold cross validation, and parameter tuning of the confidence threshold for pruning (C) and the minimum number of examples per leaf (M). The default parameters were used in the baseline case of attribute set 1 on its own. The paired-shapes results for question-answer can be viewed as highly encouraging, even more so than the critical reasoning variable at the shape level, with the highest Kappa achieved 0.87 when applying the PART algorithm to the attribute sets 1, 2, and 3. The difference between the baseline and this Kappa is significant ($p = 0.00001$) using a t-

test. Results for the other paired-shapes level variables were generally not as good as the results shown in Table 4, with the problem of too few labels leading to Kappa values of 0 in the worst cases. On the other hand, inclusion of the process-related and shape-level annotation attributes improved the learning results in virtually all cases.

Table 4: ML Results for the Question-Answer Variable running PART algorithm over 21 Maps

Att. Sets	Parameters	True Pos. Rate	True Neg. Rate	Acc.	Kappa
1 (baseline)	C = 0.25, M = 2 (Default)	59%	95%	84%	**0.59**
1	C = 0.3, M = 12 (Tuning)	67%	97%	88%	**0.69**
1 + 2	C = 0.2, M = 2 (Tuning)	88%	95%	93%	**0.83**
1 + 3	C = 0.3, M = 2 (Tuning)	92%	94%	94%	**0.85**
1 + 2 + 3	C = 0.8, M = 2 (Tuning)	92%	96%	95%	**0.87**

6. Discussion

In sum, our initial results, while preliminary, provide promise that we will be able to support moderation in ARGUNAUT, at least for some variables.

As discussed with respect to the shape-level analysis, the addition of text analysis attributes to basic structural attributes improved learning significantly. The benefits that TagHelper provided over learning with structural attributes alone, or learning with structural attributes combined with simple word attributes, shows that the kinds of linguistic features TagHelper can extract are important for distinguishing between different forms of conversational behavior. In fact, it may be that the textual features alone, without the structural features, is sufficient for training the classifiers. Upon one reviewer's suggestion, we tested this and got a Kappa of 0.67, marginally below the best Kappa of 0.68 in Table 3. So an important next step is to investigate more fully the relative contributions of the two types of attributes. Although we have not yet applied TagHelper to the paired-shapes level of analysis – it seemed less a fit for those instances and our initial results were already quite good – we will investigate whether we can improve those results, too, using TagHelper.

The paired-shapes analysis illustrated the importance of learning with structural and process attributes, both of which are critical attribute types derivable from our e-discussion maps. Including these types of attributes with the basic attributes (i.e., adding attribute sets 2 and/or 3 to attribute set 1), along with parameter tuning, led to statistically significant improved learning. In addition, part of our vision for using machine learning is to see how much we can leverage smaller building blocks, for instance the classification of individual shapes, in analyzing and learning from larger portions of the maps. We took an initial step in this direction by including individual shape annotations as attributes in training at the paired-shapes level and this supported the best results of all our experiments.

An issue related to language analysis is the translation that we did from Hebrew to English to perform the experiments described in this paper. While this was a necessary step for initial experimentation, it is an unrealistic approach if the goal is ultimately to classify Hebrew (or other language) maps. We will need to include some form of cross-language analysis in our approach. This is an issue we will investigate with respect to our TagHelper work and collaboration with Carolyn Rosé. In the short-term, however, we will continue to focus on maps translated to or originally created in English.

Our initial analysis uncovered other issues to address. For instance, the lack of positive or negative labels for a number of variables, both at the shape and paired-shapes level, led to poor results for those variables. We plan to address this in two ways. First, we will find and annotate more examples of the minority labels in our large repository of Digalo maps. Second, we will investigate the use of cost-sensitive

machine learning techniques, an approach sometimes used to deal with imbalanced data sets [13, 14]. Since in many cases the minority label is the one our moderators will be most interested in (e.g., it is more important to correctly identify when students are off than on topic), such an approach makes sense in our case.

Finally, we simply need to do more extensive machine learning experimentation to verify the viability of this approach. The current sample of data is relatively small and focuses on a limited set of e-discussions. Annotating and training on a larger corpus will provide a clearer picture of whether this direction is fruitful and will allow us to generalize the classifiers.

7. Conclusion

The ARGUNAUT system aims to provide a moderator (or teacher) with tools that will allow him or her to moderate the simultaneous e-discussions of many groups of students as they discuss, debate, and argue difficult topics using a visual argumentation software tool. Awareness Indicators, provided as part of a Moderator's Interface, are intended to alert the moderator of important events in the e-discussion, such as students not using critical reasoning in their contributions. In this paper we have discussed preliminary steps taken in using machine learning techniques to support the Awareness Indicators. Our preliminary results are quite encouraging, but there is still much work to be done. We are only in the first year of a three-year project.

References

[1] Dillenbourg, P., Baker, M., Blaye, A., & O'Malley, C. (1995). The evolution of research on collaborative learning. In P. Reimann & H. Spada (Eds.), Learning in humans and machines: Towards an interdisciplinary learning science, 189-211. Oxford: Elsevier/Pergamon.

[2] O'Donnell, A. M. (1999). Structuring dyadic interaction through scripted cooperation. In A. M. O'Donnell & A. King (Eds.), Cognitive perspectives on peer learning, 179-196. Mahwah, NJ: Erlbaum.

[3] Diziol, D., Rummel, N., Spada, H. & McLaren, B. M. (in press). Promoting Learning in Mathematics: Script Support for Collaborative Problem Solving with the Cognitive Tutor Algebra. To appear in *Computer Supported Collaborative Learning* (*CSCL-07*).

[4] Constantino-Gonzalez & Suthers (2002). Coaching Collaboration in a Computer-Mediated Learning Environment. In the *Proceedings of Computer Supported Collaborative Learning (CSCL-02)*.

[5] Vizcaíno, A. (2005). A Simulated Student Can Improve Collaborative Learning. *Journal of Artificial Intelligence in Education*, 15, 3-40.

[6] De Groot, R., Drachman, R., Hever, R., Schwarz, B., Hoppe, U., Harrer, A., De Laat, M., Wegerif, R. McLaren, B. M. & Baurens, B. (in press). Computer-Supported Moderation of E-Discussions: the ARGUNAUT Approach. To appear in *Computer Supported Collaborative Learning* (*CSCL-07*).

[7] Witten, I. H. & Frank, E. (2005). Data Mining: Practical Machine Learning Tools and Techniques, Second Edition, Morgan Kaufmann.

[8] Schwarz, B. B. & Glassner, A. (in press). The role of CSCL Argumentative Environments for Broadening and Deepening Understanding of the Space of Debate. In R. Saljo (Ed.), Information Technologies and Transformation of Knowledge.

[9] Wegerif, R. (2006). A dialogic understanding of the relationship between CSCL and teaching thinking skills. *Int'l Journal of Computer Supported Collaborative Learning*, 1(1), 143-157.

[10] Donmez, P., Rosé, C. P., Stegmann, K., Weinberger, A., & Fischer, F. (2005). Supporting CSCL with Automatic Corpus Analysis Technology. In the *Proc. of Computer Supported Collaborative Learning* (*CSCL-05*), 1-10.

[11] Mierswa, I., Wurst, M., Klinkenberg, R., Scholz, M. & Euler, T. (2006). YALE: Rapid Prototyping for Complex Data Mining Tasks. In *Proc. of the 12th ACM SIGKDD Int'l Conference on Knowledge Discovery and Data Mining (KDD 2006)*, 935-940. ACM Press.

[12] Van Rijsbergen, C. J. (1979). Information Retrieval. Butterworths, London, 2nd edition.

[13] Zadrozny, B., Langford, J. & Abe, N. (2003). A Simple Method for Cost-Sensitive Learning. IBM Technical Report RC22666, December 2002.

[14] Japkowicz, N. & Stephen, S. (2002). The Class Imbalance Problem: A Systematic Study, *Intelligent Data Analysis*, 6 (5), 429-450.

Linguistics and Language Technology, Tutorial Dialogues

This page intentionally left blank

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Analyzing the Coherence and Cohesion in Human Tutorial Dialogues when Learning with Hypermedia

Roger AZEVEDO and Moongee JEON
Dept. of Psychology, Institute for Intelligent Systems, University of Memphis
3693 Norriswood Avenue, Memphis, TN, 38152, USA
{razevedo@memphis.edu; mjeon1@memphis.edu}

Abstract. We examined the coherence and cohesion from 38 think-aloud transcriptions from a human tutorial dialogue study examining the role of tutoring on college students' learning about the circulatory system with hypermedia. The corpus we examined had a total of 800 pages. We used Coh-Metrix, a web-based tool designed to examine text and discourse, to evaluate the coherence and cohesion of the text produced between a human tutor and low-domain knowledge college students during 38 tutoring sessions. Our findings demonstrated that there were significant differences in the tutorial dialogues of the high-jump students (i.e., those who showed conceptual gains in their pretest-posttest mental models of the circulatory system) versus the medium-jumpers or no-jumpers in the semantic/conceptual overlap, the negative additive connectives incidence scores, the number of turns, and the average words per sentence. However, the tutorial dialogues of the high jump students substantially shared the syntactic and linguistic similarities with the tutorial dialogues of the medium or no jump students in the standard readability formulas, the co-referential cohesion (argument and stem overlaps), and the incidence scores of all the connectives. We argue that the semantic/conceptual overlap of the tutorial dialogues primarily promoted the high "jumps" or improvement in students' mental model and deep learning while interacting with a tutor. Our findings have implications for the design to intelligent tutorial dialogue hypermedia systems aimed at fostering learners' understanding of complex and challenging science topics.

Introduction: Cohesion and Coherence in Tutorial Dialogues

Cohesion and coherence are critical ingredients that influence text comprehension. Specifically, coherence refers to characteristics of people's mental or situation models [4, 5, 6]. For example, readers normally attempt to construct a coherent mental model while reading narrative or scientific texts [7, 8]. Cohesion, in contrast, indicates features of the texts or the textbooks that they read [4, 5, 6]. Recent studies have demonstrated that cohesion and coherence are important attributes for determining whether inferences are needed to combine text constituents [7, 8, 15], to connect the constituents with people's

prior background knowledge [14], and to establish mental or situation model in their mind [3, 7, 12, 13]. The mental model constructed by people facilitates a deeper understanding of texts and primarily depends on people's prior knowledge. Thus, the coherent mental representation established by text-based information together with people's prior knowledge is cardinal for deeper processing and understanding of texts.

These theoretical frameworks and recent progress in computational linguistics and natural language technologies have made it possible to develop a Web-based software tool called Coh-Metrix (http://cohmetrix.memphis.edu/) which assesses cohesion and coherence of text, discourse, and tutorial dialogue on the basis of over 500 measures [4]. Indices of Coh-Metrix embrace co-referential cohesion (argument overlap and stem overlap), LSA (latent semantic analysis: semantic/conceptual overlap), causal cohesion, connectives, logical operators, and other measures. Coh-Metrix computes various types of language and cohesion characteristics unlike standard readability formulas (e.g., Flesch Reading Ease or Flesch-Kincaid Grade Level) which focus primarily on word length and sentence length.

Coh-Metrix has been used extensively by our research group. For instance, Jeon and Graesser (2006) have recently used the Coh-Metrix tool to compare cohesion and readability of the tutorial dialogue of high-knowledge students who had taken physics in a college class versus low-knowledge students with no college physics background knowledge while interacting with an animated pedagogical agent called AutoTutor [9]. AutoTutor teaches college students conceptual physics and scaffolds their learning by holding conversations in natural language [9]. Their results indicated that the tutorial dialogues of the high-knowledge students were connected more cohesively and coherently than of the low-knowledge students, but there were no differences for the standard readability formulas.

In the context of this paper, Coh-Metrix was used in this human tutoring study to compare the cohesion, coherence, and readability of the tutorial dialogues of 38 non-science college students who showed high "jumps" or improvement in their mental models (from pretest to posttest) versus of students who showed medium or no jump in their mental models.

2. Method

This study reports on a subset of the data from a human tutoring study recently conducted by Azevedo and colleagues [2] of college students being tutored by a human tutor while using a hypermedia environment to learn about the circulatory system.

2.1 Tutoring Sessions.

The methodology used extensively by Azevedo and colleagues [1, 2, 10, 16] to assess students' self-regulated learning (SRL) and externally regulated learning (ERL) about science with hypermedia was used in this study. It involved the following steps: (1) obtaining informed consent; (2) administering participant questionnaire; (3) administering pretest (20 minutes); (4) providing instructions for the learning task; (5) providing participants with a tour of the hypermedia environment including its features and content;

(6) giving participants a think-aloud practice task; (7) informing participants of the overall learning goal (*"Make sure you learn about the different parts and their purpose, how they work both individually and together, and how they support the human body"*) as part of their instructions for learning about the circulatory system. During the 40-minute hypermedia learning task, the participants had access to the think-aloud instructions and the overall learning goal. Participants were free to take notes or make drawings during the learning session, although not all chose to do so. All participants were given 40 minutes to use the hypermedia environment to learn about the circulatory system. However, only the participants in the ERL (externally regulated learning) tutoring condition had access to a human tutor who would help them learn about the circulatory system by providing external regulation by prompting them throughout the session to activate their prior knowledge, create learning goals, monitor several aspects of their learning and task conditions, deploy key learning strategies (e.g., summarize, coordinate informational sources), and handle task difficulties and demands (e.g., engage in help-seeking behavior). In this study, we only examined the tutorial dialogues of the students who participated in the ERL tutoring condition to analyze the cohesion and coherence of the human tutorial dialogues.

2.2 Context and Mental Model Categorizations.

The original sample consisted of 41 non-science college students of the ERL (externally regulated learning) tutoring condition from the University of Maryland who participated for course credit. Their average age was 21 years and they were enrolled in an educational psychology course. Of the 41 students in the tutoring condition we had to drop 3 because of poor audio quality. From the remaining 38 think-aloud transcriptions, we divided them into 3 mental model groups: 18 high jumpers, 9 medium jumpers, and 11 no jumpers. The mental model assignment was based on the application of Azevedo and colleagues' mental model categories. Briefly, their scheme consists of 12 mental model categories which represent the progression from no understanding to the most accurate understanding (see Azevedo et al., 2005 for a complete description of the necessary features for each of the 12 mental models). Given the categorical nature of the mental models, we further re-categorized the 12 mental models into 3 categories: (1) *High mental model shift* includes a pretest score between 1-5 and a posttest score between 9-12; (2) *Intermediate mental model shift* includes a pretest score between 1-5 and a posttest score with 7 or 8; and (3) *No jump group* which includes anyone whose pretest mental model score is identical to their posttest mental model score (e.g., mental model of 2 on pretest and 2 on the posttest).

2.3 Materials and Procedures for Analyzing Tutorial Dialogue Data.

We collected 38 ERL (externally regulated learning) transcription logs from the 38 college students who participated in this human tutoring study. All the transcription logs were created using Microsoft Office Word files. Here is an example of the tutor-student dialogue of the ERL transcriptions:

Tutor: yeah once you, it will take you back to that page and if you scroll down you'll see it. It's the one, yeah.
Student: ok.
AUDIO VIDEO: … There are two stages in each heartbeat cycle-the systole and the diastole…
Tutor: Okay, so the oracles are the same thing as the atrium
Student: Ok, I was going to ask you about that. So now I'm gonna learn about arteries and capillaries and what the differences are between them.
Tutor: ok.
Student:
READS: Arteries have thicker walls than veins to withstand the pressure of blood being pumped from the heart…

For the analyses of the tutorial dialogues with Coh-Metrix, we extracted only tutor and student turns in the transcriptions, excluding AUDIO VIDEO and READS parts of the transcriptions. Next, we separated the tutor-student tutorial dialogues into the tutor and student turns because the main research question of this study was *how the cohesion, coherence, and readability of the tutorial dialogues of students who showed high "jump" or improvement in their mental models compare with students who showed medium or no jump in their mental models.* We used Coh-Metrix entering the tutor-student turns, the tutor turns, and the student turns. However, we only report the analyses of the students' dialogues for this study.

3. Results

In this section, we focus on a few specific Coh-Metrix indices. More specifically, we selected the indices such as the LSA (latent semantic analysis), the co-referential overlap scores, the standard readability formulas, the incidence scores of connectives, the number of turns, and the average words per sentence because the main research interest of this paper was to compare the cohesion, coherence, and readability of the tutorial dialogues among the groups (High jumpers versus Medium jumpers or No jumpers). The following results presented in this section were based on several one-way ANOVAs conducted between the means of several Coh-Metrix indices of the three mental model jump groups.

3.1 Analysis of LSA (Latent Semantic Analysis) Cosines of Adjacent Sentences.

The LSA cosines of adjacent sentence to sentence indicate the semantic/conceptual overlap of the adjacent sentences. High LSA cosines of adjacent sentence to sentence indicated that the adjacent sentences in the dialogues are linked more semantically and coherently. We performed an ANOVA on the LSA cosines of adjacent sentence to sentence as a function of students' jump levels for their mental model (High Jumpers, Medium Jumpers, and No Jumpers). Our results showed that there were significant differences among the different jumper groups for the LSA cosines of adjacent sentence to sentence, F (2, 35) = 5.97, MSE = .002, $p < .05$.

Table 1. Means (and SDs) for the indices of Coh-Metrix by mental model jump group conditions

	Mental model jump groups		
	High Jumpers	Medium Jumpers	No Jumpers
	Mean (SD)	*Mean (SD)*	*Mean (SD)*
LSA cosines of adjacent sentences	.21 (.06)	.15 (.04)	.17 (.03)
LSA cosines of turn to turn	.17 (.06)	.12 (.02)	.14 (.03)
Negative additive connectives incidence score	4.7 (2.4)	5.4 (1.8)	7.7 (4.1)
Number of turns	201 (55)	163 (46)	148 (42)
Average words per sentence	4.89 (1.19)	6.31 (2.29)	6.79 (2.99)
Flesch Reading Ease	81.9 (6.7)	84.6 (5.4)	83.9 (8)
Flesch-Kincaid Grade Level	3.1 (1)	3 (1.2)	3.3 (1.8)
Argument overlap of adjacent sentences	.13 (.09)	.14 (.07)	.17 (.09)
Stem overlap of adjacent sentences	.13 (.09)	.1 (.07)	.13 (.09)
All connectives incidence score	89.9 (18.4)	91.3 (17.7)	87 (21.7)

We then performed post hoc LSD (least significant difference) pairwise comparisons to clarify which jumper groups were significantly different. Post hoc LSD pairwise comparisons showed that the mean of the LSA cosines of adjacent sentence to sentence of the tutorial dialogues of the high jumpers (M = .21) were significantly higher than of the medium (M = .15) and no jumpers (M = .17), each p < .05, but the medium and no jumpers did not significantly differ from each other (see Table 1). The results suggest that the coherence (means local coherence) of adjacent sentences of the tutorial dialogue of the high jumpers was higher than the medium and no jumpers.

3.2 Analysis of LSA Turn to Turn.

The LSA cosines of turn to turn indicate the semantic/conceptual overlap of the turns in the dialogues. An ANOVA was performed on the LSA cosines of turn to turn with jump levels of students (High Jumpers, Medium Jumpers, and No Jumpers). There were significant differences among the groups, F (2, 35) = 4.12, MSE = .002, p < .05, and the follow-up post hoc LSD pairwise comparisons indicated that the mean LSA cosines of turn to turn of the dialogues of the high jumpers (M = .17) were higher than of the medium (M = .12) and the no jumpers (M = .14), each p < .05 (see Table 1). The results indicate that the coherence (means global coherence) of turn to turn of the tutorial dialogue of the high jumpers was higher than of the medium and no jumpers.

3.3 Negative Additive Connectives Incidence Scores.

The incidence scores of the negative additive connectives in the dialogues of students were calculated using Coh-Metrix. An incidence score is defined as the number of occurrences

of a particular category per 1,000 words. We performed an ANOVA on the negative additive connectives (e.g., *however* and *but*) as a function of jump levels of students (High Jumpers, Medium Jumpers, and No Jumpers). The results showed that there were significant differences among the groups, F (2, 35) = 3.65, MSE = 8.53, $p < .05$ and the post hoc LSD pairwise comparisons represented that the negative additive connectives incidence scores of no jumpers (M = 7.7) were higher than the high (M = 4.7) jumpers, $p < .05$. These results suggest that the high jumpers tended to use less negative additive connectives than no jumpers while interacting with a tutor indicating that the negative additive connectives had a negative effect on learning. However, there was no significant effect among the group for the incidence score of all the connectives (see Table 1).

3.4 Number of Turns and Average Words per Sentence.

The number of turns and the average words per sentence in the students' dialogues were calculated using Coh-Metrix. ANOVAs were performed on the number of turns and the average words per sentence with jump levels of students (High Jumpers, Medium Jumpers, and No Jumpers). Our results showed that there were significant differences among the three groups for the number of turns, F (2, 35) = 4.25, MSE = 2446.9, $p < .05$, and the average words per sentence, F (2, 35) = 3.16, MSE = 4.45, $p = .055$. Follow-up post hoc LSD pairwise comparisons revealed that the number of turns of high jumpers (M = 201) was significantly higher than no jumpers (M = 148), $p < .05$, whereas the average words per sentence of the higher jumpers (M = 4.89) were significantly lower than no jumpers (M = 6.79), $p < .05$. These results indicate that the high jumpers tended to interact with a tutor longer than no jumpers. However, the sentences of the high jumpers were more succinct than the sentences of no jumpers; more succinct, more learning indicating that the high jumpers tended to use shorter sentences than no jumpers to verify concepts and information of the subject matter while interplaying with a tutor.

3.5 Analyses of Standard Readability Formulas.

The Flesch Reading Ease score is a number ranging from 0 to 100, indicating that a higher score represents easier reading, whereas the Flesch-Kincaid Grade Level ranging 0 to 12 converts the Flesch Reading Ease Score to a U.S. grade-school level indicating that a higher score means more difficult reading. Two ANOVAs were conducted on the two standard readability formulas as a function of jump levels of students (High Jumpers, Medium Jumpers, and No Jumpers). The results showed that there were no significant differences among the groups for both. These results suggest that the standard readability formulas were not sensitive to extract semantic/conceptual meanings from the tutorial dialogues. Table 1 illustrates the results of this analysis.

3.6 Analyses of Adjacent Argument and Stem Overlap.

The adjacent argument overlap is the proportion of adjacent sentences that share one or more arguments (e.g., noun, pronoun, noun-phrase) and the stem overlap is the proportion

of adjacent sentences that share one or more word stems (e.g., **swim**, **swim**mer, **swim**ming). Two ANOVAs were performed on the adjacent argument and stem overlap scores with jump levels of students (High Jumpers, Medium Jumpers, and No Jumpers), but there were no significant differences between the groups for both (see Table 1). These results suggest that the co-referential overlap scores of the tutorial dialogues were not reliable indicators to predict deep learning of students while they interacting with a tutor.

4. Summary of Results

In sum, we examined the coherence and cohesion from 38 think-aloud transcriptions (800 pages) from a human tutorial dialogue study examining the role of tutoring on college students' learning about the challenging and complex science topic with hypermedia. We used Coh-Metrix to evaluate the coherence and cohesion of the text produced between a human tutor and low-domain knowledge college students during 38 tutoring sessions. Our findings demonstrated that there were significant differences in the tutorial dialogues of the high jump students versus the medium or no jump students in the semantic/conceptual overlap, the negative additive connectives incidence score, the number of turns, and the average words per sentence. However, the tutorial dialogues of the high jump students substantially shared the syntactic and linguistic similarities with the tutorial dialogues of the medium or no jump students in the standard readability formulas, the co-referential cohesion (argument and stem overlaps), and the incidence scores of all the connectives. We suspect that the semantic/conceptual overlap of the tutorial dialogues primarily promoted the high "jumps" or improvement in students' mental model and deep learning while interacting with a tutor. Given the novelty of these methods within the AI Ed community we argue that our findings have implications for the design to intelligent tutorial dialogue hypermedia systems aimed at fostering learners' understanding of complex and challenging science topics.

5. Acknowledgements

This research was supported by funding from the National Science Foundation (REC#0133346 and REC#0633918) awarded to the first author. The authors would like the thank Jeffrey Greene, Daniel Moos, Fielding Winters, and Neil Hofman for assistance with data collection and transcribing the audio data. We also thank Jennifer Scott for preparing the transcriptions prior to using CohMetrix and Amy Witherspoon for comments on a previous draft.

References

[1] Azevedo, R., Cromley, J., Winters, F., Moos, D., & Greene, J. (2005). Adaptive human scaffolding facilitates adolescents' self-regulated learning with hypermedia. *Instructional Science, 33,* 381-412.

[2] Azevedo, R., Moos, D., Greene, J., Winters, F., & Cromley, J. (in press). Why is externally-regulated learning more effective than self-regulated learning with hypermedia? *Educational Technology Research and Development*.
[3] Gernsbacher, M. (1997). Two Decades of Structure Building. *Discourse Processes, 23,* 265-304.
[4] Graesser, A., McNamara, D., Louwerse, M., & Cai, Z. (2004). Coh-Metrix: Analysis of text on cohesion and language. *Behavioral Research Methods, Instruments, and Computers, 36,* 193-202.
[5] Graesser, A., Louwerse, M., McNamara, D., Olney, A., Cai, Z., & Mitchell, H. (in press). Inference generation and cohesion in the construction of situation models: Some connections with computational linguistics. In F. Schmalhofer and C. Perfetti (Eds.), *Higher level language processes in the brain: Inferences and comprehension processes*. Mahwah, NJ: Erlbaum.
[6] Graesser, A., McNamara, D., & Louwerse, M. (2003). What do readers need to learn in order to process coherence relations in narrative and expository text. In A.P. Sweet & C.E. Snow (Eds.), *Rethinking reading comprehension* (pp. 82-98). New York, NY: Guilford Publications.
[7] Graesser, A., Singer, M., & Trabasso, T. (1994). Constructing inferences during narrative text comprehension. *Psychological Review, 101,* 371-395.
[8] Graesser, A., & Bertus, E. (1998). The construction of causal inferences while reading expository texts on science and technology. *Scientific Studies of Reading, 2,* 247-269.
[9] Graesser, A., Person, N., Lu, Z., Jeon, M-G., & McDaniel, B. (2005). Learning while holding a conversation with a computer. In L. PytlikZillig, M. Bodvarsson, & R. Bruning (Eds.), *Technology-based education: Bringing researchers and practitioners together* (pp. 143-167). Greenwich, CT: Information Age Publishing.
[10] Greene, J., Moos, D., Azevedo, R., & Winters, F. (in press). Exploring differences between gifted and grade-level students' use of self-regulatory learning processes with hypermedia. *Computers & Education*.
[11] Jeon, M-G., & Graesser, A. (November, 2006). *Using Coh-Metrix to Assess the Role of Background Knowledge on Cohesion in Tutorial Dialogue*. Symposium presentation at the 36th Annual Meeting of the Society for Computers in Psychology. Houston, TX.
[12] Kintsch, W. (1988). The Role of Knowledge in Discourse Comprehension: A Construction Integration Model. *Psychological Review, 95,* 163-182.
[13] Kintsch, W. (1998). *Comprehension: A paradigm for cognition*. New York: Cambridge University Press.
[14] McNamara, D., Kintsch, E., Songer, N., & Kintsch, W. (1996). Are good texts always better? Text coherence, background knowledge, and levels of understanding in learning from text. *Cognition and Instruction, 14,* 1-43.
[15] Millis, K., & Graesser, A. (1994). The time-course of constructing knowledge-based inferences for scientific texts. *Journal of Memory and Language, 33,* 583-599.
[16] Moos, D., & Azevedo, R. (2006). The role of goal structure in undergraduates' use of self-regulatory variables in two hypermedia learning tasks. *Journal of Educational Multimedia and Hypermedia, 15*, 49-86.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Beyond the *code-and-count* analysis of tutoring dialogues [1]

Stellan OHLSSON [a], Barbara DI EUGENIO [a,2], Bettina CHOW [a], Davide FOSSATI [a], Xin LU [a], and Trina C. KERSHAW [b]

[a] *University of Illinois at Chicago, IL, USA*
[b] *University of Massachusetts at Dartmouth, MA, USA*

Abstract. In this paper, we raise a methodological issue concerning the empirical analysis of tutoring dialogues: The frequencies of tutoring moves do not necessarily reveal their causal efficacy. We propose to develop coding schemes that are better informed by theories of learning; stop equating higher frequencies of tutoring moves with effectiveness; and replace ANOVAs and chi-squares with multiple regression. As motivation for our proposal, we will present an initial analysis of tutoring dialogues, in the domain of introductory Computer Science.

Keywords. Tutoring dialogues, annotation, multiple regression

1. Introduction

In this paper, we raise a methodological issue concerning the empirical analysis of tutoring dialogues, in general and in the service of building dialogue interfaces to Intelligent Tutoring Systems (ITSs). Like many others [2,7,8,10,13,19], we have been engaged in the collection and analysis of one-on-one tutoring dialogues [4,5]. The goal is to glean insight into why tutoring is effective, and to build computational models of effective tutoring strategies, which in turn are an essential component of dialogue interfaces to ITSs. However, even with so much effort, the community does not yet agree on a repertoire of effective tutoring strategies. We believe this is because everybody, including ourselves, has been equating effectiveness with frequency: namely, what tutors, in particular expert tutors, do most often is what is deemed to be effective. However, frequency per se does not prove effectiveness, as we will explicate below. In addition, coding categories for tutoring dialogues have often been developed bottom-up, certainly informed by linguistics theories of speech acts, but not directly informed by theories of learning. In this paper, we present our argument, and we follow up with two proposals: develop coding schemes that are better informed by theories of learning, and replace what we call the *code-and-count* methodology with an analysis based on multiple regression. As motivation for our proposal, we will present an initial analysis of tutoring dialogues, in the domain of introductory data structure and algorithms in Computer Science.

[1]This work is funded by awards N000140010640 and N000140710040 from the Office of Naval Research, and in part by awards IIS–0133123 and IIS–0513553 from the National Science Foundation.
[2]Corresponding Author: Computer Science (M/C 152), 851 S. Morgan St., Chicago, IL 60607, USA.

2. The analysis of tutoring dialogues: A critique

2.1. Coding Categories

Tutor-tutee interactions are so rich and complex that researchers have not yet converged on a shared set of tutoring moves, i.e., the basic units of analysis. For example, Person [16] lists 23 categories of tutoring behaviors, including *hint, prompt, pump, bridge, summarize, ask clarification questions, ask comprehension-gauging questions, provide counterexamples, give direct instruction, force a choice, provide a preview, provide examples, complain, revoice*. Although long, this list is not inclusive. For example, VanLehn et al. [19] coded tutoring transcripts for the occurrence of *impasses*, *explanations* and *hints about subgoals*; the first and third of these are not identical to any category in Person's list. The coding categories from [7] overlap with those mentioned so far, e.g. hints, and so do our own, e.g., prompting, [5], but do not coincide. The point is that tutoring research has not converged on a widely agreed-upon theory of how the behavior of tutors should be segmented and characterized, a necessary preliminary step to investigating which parts of tutoring produce the high learning gains. The various lists of tutoring moves have sometimes been created in a bottom-up fashion, in response to the contents of the transcripts collected by a research team. Another influence is to be found in linguistic theory, especially theories of dialogue acts, and yet another in prior pedagogical concepts about how tutoring might work (e.g., Socratic questioning). To date, attempts to answer the question of why tutoring is effective have been less responsive to what we know about how people learn. Yet, instruction can only work by supporting learning, so considering how people learn is a natural starting point [14]; a core coding scheme of this sort can be later enriched with other categories, i.e. those suggested by theories of dialogue acts, if necessary.

2.2. *The* code-and-count *methodology*

The question of how tutoring strategies and moves should be conceptualized interacts with the standard methodology used in empirical tutoring studies. The latter often proceed by classifying tutoring behaviors and then assessing which of those behaviors are the most important in achieving the superior learning outcomes associated with tutoring by counting the frequency of each behavior in a corpus of transcripts. This code-and-count methodology can only be as informative as the coding categories; if the latter do not categorize tutoring moves in a way that corresponds to the actual workings of tutoring, then the frequencies of those categories will be uninformative.

Even if a code-and-count study began with a principled set of categories, there are other methodological problems that blunt the impact of such studies. It rests on the assumption that the tutoring behaviors that account for the superior learning gains obtained with tutoring would turn out to be those behaviors in which the tutors engage frequently. Hence, code-and-count would generate the right tutoring theory by revealing the highest-frequency tutoring moves. In retrospect, this assumption (made by everybody in the community, including ourselves) appears to us to be flawed. There is no guarantee that the moves that account for most of the variance in learning outcomes are necessarily the moves that occur most frequently. After all, the tutors themselves, even expert tutors, do not have a theory of tutoring (or else we could accomplish our objective by asking them)

– even if some of them, for example CIRCSIM experienced physiology tutors, constantly reflect on and evaluate their own tutoring [12]. Furthermore, human interactions are very complicated and shaped by multiple factors, including the standard interaction patterns of the surrounding culture and the degree and nature of the rapport between a particular tutor and a particular tutee. Hence, the maximally effective tutoring moves – the ones most worthwhile to mimic in an artificial tutoring system – might be few and embedded within a lengthy interaction that is rich, varied and complicated for a variety of reasons that bear little causal relation to the learning outcome. Compare the situation to negotiation dialogues: If two negotiators after a lengthy discussion reach a particular agreement, which utterances in their transcript were causally related to the outcome? Clearly, it is not necessarily the type of utterance that was most frequent. A more sophisticated approach is to code-and-count tutoring dialogues produced by expert tutors, specifically. It is reasonable to assume that expert tutors will do more of whatever it is that works in tutoring, so the most frequent types of tutoring moves by expert tutors should have a high probability of cutting close to the tutorial bone. However, even a study of expert tutors can only be as revealing as the set of categories is insightful, and there is no guarantee that the highest frequency moves are those with the strongest impact on learning.

In addition, studies of expert tutors have weaknesses of their own. As Person recently documented [16], only a few expert tutors have been studied [3,7,11], in part because the total set of studies of expert tutors is small and in part due to the fact that the very same expert tutors have appeared in more than one study. In addition, tutors are often identified as 'expert' on the basis of indirect indicators such as how long they have been tutoring, how much they are paid, etc. With respect to the goal of identifying the dimensions of tutoring that are causally related to high learning gains, the operative concept is not *expert* but *effective* tutors, and the measure of effectiveness is some measure of learning outcomes, not any property of the tutors themselves. If some of the expert tutors that have been studied do not in fact achieve high learning gains, their presence in the data base constitutes noise. In the beginning of our research on expert vs. non expert tutors [5,6] we thought to overcome such weaknesses by (a) verifying that our expert tutors did indeed produce higher learning gains by using pre- and posttests, and (b) by looking at *differences* between more and less effective tutors. In other words, we thought that the differential frequencies that would turn up in the code-and-count would indicate whatever it is that effective tutors do more often than ineffective tutors, and those types of behaviors would be good candidates for the tutoring moves that produce the better learning outcomes. This same approach was taken by the CIRCSIM-Tutor group [7,9]. Although we believe that these two methodological improvements – measure learning gains and look at differential rather than absolute frequencies – are important, we now recognize that they do not completely overcome the weaknesses that we have identified above: the creation of a coding system that does not take into account theories of learning and the assumption that the effective tutoring moves are necessarily more frequent than those with less causal impact on learning. In addition, as we will discuss below, a tutor who has been deemed *expert* according to some a priori criteria may not be more effective than a tutor who has not been deemed as expert.

3. Proposed Solution

3.1. Coding Categories

The system for categorizing tutoring behavior should be informed by what we know about learning (as well as by the manifest content of the relevant tutoring transcripts). Although no theory of skill acquisition is widely accepted, most researchers would agree that people learn in at least the following four ways:

1. People can learn by capturing and encoding successful steps during problem space search.
2. People can learn by detecting and correcting their errors.
3. People can learn by encoding declarative facts about the domain or about the types of problems or tasks they are practicing, delivered via discourse.
4. People can learn by compiling declarative representations of the tactics and strategies they are trying to learn into executable cognitive strategies.

The claim is not that this is the complete theory for how people learn, only that people learn in at least these four ways. Each of these types of learning suggests a particular way in which instruction can support learning:

1. A tutor can provide positive feedback to confirm that a correct but tentative student step is in fact correct.
2. A tutor can provide negative feedback that helps a student detect and correct an error.
3. A tutor can state the declarative information about the domain.
4. A tutor can tell the student how to perform the task.

For each of these types of tutorial inputs, we can explain why and how they might support learning in terms of current cognitive theories of skill acquisition [1,15,17,18]; and they have all been documented as occurring in tutoring dialogues. It seems reasonable, then, to code tutoring transcripts for the occurrence of (at least) these four categories of tutoring behavior. Note that the empirical methodology that we propose below will signal to us whether these categories are sufficient or we need to add other categories.

3.2. Multiple regression

We propose to move away from absolute frequencies to a different criterion for how to decide whether a category of tutoring behavior is causally related to learning: Multiple regression of learning outcomes per tutoring session onto the frequencies of the different tutoring moves. First, we intend to shift the unit of analysis from *tutor* to *tutoring session*. Past research implicitly assumed that tutoring expertise is general across recipients; that is, if a tutor is effective with student X, he/she tends to be effective with student Y as well. Therefore, we can uncover the moves of effective tutoring by statistical aggregation of code-and-count data over students and sessions but within tutor. However, the richness of human dialogues once again intervenes. It is highly likely that each tutor succeeds better with some students than with others; better rapport, more closely related and matching linguistic habits or thoughts, and so forth. It is therefore reasonable to focus on tutoring sessions, not tutoring persons. The question then becomes, what happens in those sessions in which much learning happened, and what differentiates them

from those session in which it didn't? This is clearly a different question than asking what certain persons, expert tutors, tend to do that other persons do not do, or do less of. The next methodological innovation, which develops a trend that is already present in some recent tutoring studies [19], is to move away from ANOVAs and chi-squares to a correlational approach. We anticipate to see a wide variety of learning outcomes across sessions, and there may be wide variations in the frequencies of the four theory-based tutoring categories outlined above. To answer the question which of these categories are causally related to the learning outcomes, we will employ multiple regression, with the amount of learning per session as the predicted variable and the frequencies of the tutoring behaviors as the predictor variables. The beta weights, the partial regression coefficients, will provide information regarding the relative strength of the relations between the relevant tutoring moves and the learning outcomes. This method still relies on the frequency of each type of tutoring move as an important variable, but it does not assume that the more effective tutoring moves are necessarily more frequent, in absolute numbers, than the less effective ones. The method reveals whether *variation* in the frequency of one type of tutoring move is more strongly related to *variation* in learning outcomes than another type of move, regardless of which type of move is more or less frequent. An additional advantage of the correlational methodology is that it will tell us if we are on the wrong track: One possible outcome is that none of the partial regression coefficients are significant. This is a sign that the categories of tutoring moves are not the right ones, and that the transcripts need to be re-coded with a different set of categories (i.e., that the theory behind the category system is wrong and needs to be replaced by a different learning theory). There is no counterpart to this in the code-and-count methodology: Any set of codes will always generate some frequencies, and there will always be some code that turns out to be more frequent than another. There is no built-in warning signal that the category system is fundamentally flawed, but in the correlational approach, low and non-significant correlations do provide such a warning.

4. An initial analysis of tutoring dialogues in Computer Science

Our current goal is to investigate computational models of tutoring dialogues in order to build a dialogue interface to an ITS for basic data structures and algorithms. We have thus engaged in an extensive collection of Computer Science tutoring dialogues. At the moment, we have data from 82 subjects altogether, 28 control and 54 tutored. At the time of writing, we have transcribed 80% of the dialogues and started to code those already transcribed. Hence, we are not in a position to report a complete multiple regression analysis. However, we include here some preliminary findings that show the need for this kind of analysis. The subjects were recruited from CS introductory courses. They were either (a) tutored by a very experienced tutor (CS professor with 30 years experience in small liberal art schools), (b) tutored by a less experienced tutor (a CS senior with just few hours of "helping friends" under his belt), or (c) not tutored (control condition). They were given a 15 minute pretest immediately before the tutoring session, and the same posttest immediately afterward. The test contains eight problems, divided into three groups: problems 1 and 2 on linked lists, problems 3 and 4 on stacks, and problems 5-8 on binary search trees (BSTs). The students in the control condition took the same tests but in a classroom setting; instead of being tutored, they read appropriate excerpts of

the textbook, at their own pace. The tutoring sessions lasted at most 40 minutes, with some advanced students finishing a bit earlier. The pre- and posttests were graded by two graders who were blind to condition. Each test problem was worth 5 points, for a total of 40 points. Grader differences of 2 points or more were resolved by discussion, smaller grader differences by averaging over the two graders. The measure of learning gain was the difference between the post- and pretest scores. Initial general findings on learning within groups are that (these results are based on paired sample t-tests, where the two variables of interest are the scores on the pre- and posttest):

- students learn significantly in all conditions. Specifically, both CS majors and non CS majors learn significantly in each problem
- students in the control condition learn significantly on half of the problems, i.e. problems 3, 4 (stacks), and 5, 6 (BSTs)
- students with the less experienced tutor learn significantly on 5 problems out of 8, i.e. problems 3, 4 (stacks), and 5, 6, 7 (BSTs)
- students with the more experienced tutor learn significantly on each problem

Further, comparing students across different conditions, we find that (all these results are based on ANOVAs, followed by Bonferroni tests):

- both tutors are effective, i.e., subjects tutored by either tutor learn more than subjects in the control condition
- even if subjects learn a bit more with the more experienced tutor, there is no significant difference between the subjects tutored by one tutor or the other
- subjects who are not CS majors learn more than subjects who are CS majors
- there are differences for some selected problem regarding learning gain between tutored and control subjects, and between CS and non-CS majors

To start with our proposed methodology, we ran three multiple regressions, one for each subject matter (linked lists, stacks, binary search trees). Of concern are three potential predictor variables that are of interest in understanding what happens during tutoring: the students prior knowledge (pretest score), the level of experience of the tutor (less, more), and the time spent during tutoring on problems pertaining to each subject matter topic (time on task). We used both the posttest score and the gain score as measures of learning. We report the posttest score analysis. (The gain score analysis gave the same results.) We report the results of the multiple regression in Table 1. We found that the experience of the tutor did not significantly affect the posttest scores. More specifically, for linked lists, higher pretest scores and increased time on task predicted higher posttest scores: pretest score accounts for 31% of the variance, and time on task for 6%; for stacks, higher pretest scores predicted higher posttest scores, accounting for 50% of the variance; for trees, higher pretest scores predicted higher posttest scores, accounting for 18% of the variance.

These preliminary results suggest the following. First, we notice that the experience level of the tutor did not affect average learning gains, nor did tutor experience explain a significant portion of the variance in the learning gains on any of the three subject matter topics. This reinforces our contention that we should analyze tutoring dialogues by focusing not on the participant (i.e., tutor, or type of tutor or student), but rather, on the session as unit of analysis. Whereas the distinction between more and less expert tutors, which we subscribed to in our previous work, is an appealing one, it does not

		R^2	β	p
Linked lists	Pretest	0.308	0.635	< 0.001
	Tutor experience	0.001	0.033	ns
	Time on task	0.063	0.294	0.009
Stacks	Pretest	0.505	0.665	< 0.001
	Tutor experience	0.002	0.041	ns
	Time on task	0.006	-0.061	ns
BSTs	Pretest	0.180	0.483	< 0.001
	Tutor experience	0.078	-0.115	ns
	Time on task	0.025	0.234	ns

Table 1. Results from multiple regression, by topic

take into account the other half of the equation, i.e., the student. This observation does not exclude that a more experienced tutor is likely to have more successful sessions, and hence, be more effective on the whole, than a less experienced tutor [5,9]. Second, the strongest predictor of posttest score is the pretest score. Time on task comes in as second best. These are not surprising, but common findings in educational research. The point for present purposes is: The variation in the learning gains is due to many factors, each factor being responsible, so to speak, for a portion of that variance. The more variance is explained by one variable, the less is left to be explained by another. Any claim to the effect that such and such a tutoring move is causally related to learning gains must show that variations in the use of that tutoring move explains variance in learning gains, over and above what is explained by other variables such as prior knowledge and time on task. Unless the methodology for analyzing tutoring dialogues takes such variables into account, the empirical support in favor of a hypothesized tutoring move is weakened. Third, even after we take these variables into account, depending on topic there is still between 80% and 50% of the variance in the posttest scores left to explain. The obvious hypothesis is that this portion of the variance is indeed due to variations in how the tutor-tutee interaction went in any one tutoring session. To assess this hypothesis, the analysis needs to be repeated with the frequencies of the tutoring moves of interest as predictor variables.

5. Conclusions

To analyze tutoring dialogues with the purpose of extracting a theory of tutoring that can inform the design of ITSs is to ask which of the many and varied actions that tutors take in the course of interacting with a student is causally related to the high learning gains that attract us to the tutoring scenario in the first place. Frequency does not prove causal efficacy. We believe this is why code-and-count studies based on bottom-up coding schemes have so far failed to resolve the issue of what makes tutoring effective. We are developing an alternative methodology, and the purpose of this paper is to put that methodology before the ITS research community for comment and criticism. The methodology has the following parts: First, coding schemes should start from what is known about learning. A tutoring move should be identified in terms of which type of learning they support. Second, data analysis should use the individual tutoring session as the unit of analysis, not the tutor. Third, the analyses of tutoring dialogues should include

data from all the variables that are known to affect educational outcomes, such as prior knowledge, time on task, etc. Fourth, the variable of interest is not any property of the tutors per se such as number of years of tutoring experience, but the learning outcomes. Fifth, to prove that a tutoring move is effective is to prove that it accounts for variance in learning gains over and above the variance explained by such base line variables. One tool to accomplish this is multiple regression.

References

[1] John R. Anderson. Knowledge compilation: The general learning mechanism. In R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, editors, *Machine Learning*, volume 5, pages 289–310. Los Altos, CA: Kaufmann, 1986.

[2] Michelene T. H. Chi, Stephanie A. Siler, Takashi Yamauchi, and Robert G. Hausmann. Learning from human tutoring. *Cognitive Science*, 25:471–533, 2001.

[3] Teresa del Soldato and Benedict du Boulay. Implementation of motivational tactics in tutoring systems. *Journal of Artificial Intelligence in Education*, 6(4):337–378, 1995.

[4] Barbara Di Eugenio, Davide Fossati, Dan Yu, Susan Haller, and Michael Glass. Natural Language Generation for Intelligent Tutoring Systems: a case study. In *AIED 2005, the 12th International Conference on Artificial Intelligence in Education*, Amsterdam, The Netherlands, 2005.

[5] Barbara Di Eugenio, Trina C. Kershaw, Xin Lu, Andrew Corrigan-Halpern, and Stellan Ohlsson. Toward a computational model of expert tutoring: a first report. In *FLAIRS06, the 19th International Florida AI Research Symposium*, Melbourne Beach, FL, 2006.

[6] Barbara Di Eugenio, Xin Lu, Trina C. Kershaw, Andrew Corrigan-Halpern, and Stellan Ohlsson. Positive and negative verbal feedback for intelligent tutoring systems. In *AIED 2005, the 12th International Conference on Artificial Intelligence in Education*, Amsterdam, The Netherlands, 2005. (Poster).

[7] Martha Evens and Joel Michael. *One-on-one Tutoring by Humans and Machines*. Mahwah, NJ: Lawrence Erlbaum Associates, 2006.

[8] Barbara A. Fox. *The Human Tutorial Dialogue Project: Issues in the design of instructional systems*. Lawrence Erlbaum Associates, Hillsdale, NJ, 1993.

[9] Michael Glass, Jung Hee Kim, Martha W. Evens, Joel A. Michael, and Allen A. Rovick. Novice vs. expert tutors: A comparison of style. In *MAICS-99, Proceedings of the Tenth Midwest AI and Cognitive Science Conference*, pages 43–49, Bloomington, IN, 1999.

[10] Arthur C. Graesser, Natalie K. Person, and Joseph P. Magliano. Collaborative dialogue patterns in naturalistic one-to-one tutoring. *Applied Cognitive Psychology*, 9:495–522, 1995.

[11] M. R. Lepper, M. F. Drake, and T. O'Donnell-Johnson. Scaffolding techniques of expert human tutors. In K. Hogan and M. Pressley, editors, *Scaffolding student learning: Instructional approaches and issues*. Cambridge, MA: Brookline, 1997.

[12] Joel A. Michael and Allen A. Rovick. *Problem-solving in physiology*. Prentice Hall, Upper Saddle River, NJ, 1999.

[13] Johanna D. Moore, Benoît Lemaire, and James A. Rosenbloom. Discourse generation for instructional applications: Identifying and exploiting relevant prior explanations. *Journal of the Learning Sciences*, 5(1):49–94, 1996.

[14] Stellan Ohlsson. System hacking meets learning theory: Reflections on the goals and standards of research in artificial intelligence and education. *International Journal of Artificial Intelligence and Education*, 2(3):5–18, 1991.

[15] Stellan Ohlsson. Learning by specialization and order effects in the acquisition of cognitive skills. In E. Ritter, J. Nerb, T. O'Shea, and E. Lehtinen, editors, *In order to learn: How the sequence of topics affects learning*. New York, NY: Oxford University Press, 2007. To appear.

[16] Natalie Person. Why study expert tutors? Presentation at ONR Contractors' Conference on Instructional Strategies, February 2006.

[17] R. Sun, P. Slusarz, and C. Terry. The interaction of the explicit and the implicit in skill learning: A dual-process approach. *Psychological Review*, 112:159–192, 2005.

[18] Kurt VanLehn. Learning by explaining examples to oneself: A computational model. In S. Chipman and A. L. Meyrowitz, editors, *Foundations of knowledge acquisition*, pages 25–82. Boston: Kluwer, 1993.

[19] Kurt VanLehn, Stephanie Siler, and Chaz Murray. Why do only some events cause learning during human tutoring? *Cognition and Instruction*, 21(3):209–249, 2003.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Profiling Student Interactions in Threaded Discussions with Speech Act Classifiers

Sujith RAVI and Jihie KIM
University of Southern California/Information Sciences Institute
4676 Admiralty Way, Marina del Rey, CA 90292 USA

Abstract. On-line discussion is a popular form of web-based computer-mediated communication and is an important medium for distance education. Automatic tools for analyzing online discussions are highly desirable for better information management and assistance. This paper presents an approach for automatically profiling student interactions in on-line discussions. Using N-gram features and linear SVM, we developed “speech act” classifiers that identify the roles that individual messages play. The classifiers were used in finding messages that contain questions or answers. We then applied a set of thread analysis rules for identifying threads that may have unanswered questions and need instructor attention. We evaluated the results with three human annotators, and 70-75% of the predictions from the system were consistent with human answers.

Keywords: On-line discussion board, speech act, discussion assessment

Introduction

Web-enhanced courses and distance education courses are becoming increasingly popular. Engagement in on-line discussions is an important part of student activities in distance education. Our work was motivated by lack of flexibility and poor assistance capability of existing discussion boards, and by the potential for utilizing natural language processing (NLP) techniques in discussion thread analysis. We are developing instructional tools for phpBB, a popular open-source bulletin board with good community support [1].

In this paper we present an approach for automatically classifying patterns of student interactions on discussion boards. In particular, the system identifies discussion threads that may have unanswered questions and need instructor attention. In profiling discussion threads and classifying patterns of student interactions, we adopt the theory of Speech Acts proposed by (Austin, 1962 [2]; Searle, 1969 [3]) and define a set of speech acts (SAs) that relate messages in discussion threads. That is, discussion threads are viewed as a special case of human conversation, and each message is classified with respect to previous messages in the same thread as a *question*, *answer*, *elaboration* or/and *correction*. We use word sequence features and SVM (Support Vector Machine) algorithms [4] in developing automatic *SA classifiers*. We have developed two speech act classifiers: question classifier and answer classifier. The question classifier identifies messages that play a role of asking questions and the answer classifier detects messages with answers in response to a previous message. Given the classification of

individual messages, the *thread profiler* then applies a set of thread analysis rules for finding threads that have questions without corresponding answers. Such threads may need more attention from the instructor since students may have unanswered questions.

We applied these techniques in classifying discussion corpus from two undergraduate computer science courses on operating systems. We found that development of classifiers is very challenging because, unlike common corpora that are often used by NLP systems, discussions among undergraduate students are highly unstructured and noisy. Cleaning and preprocessing raw data, transforming them into more coherent data sets, and selecting useful features among many potential feature candidates were most challenging. The current SA classifiers that we have developed for questions and answers have accuracies of 88% and 73% respectively. The thread profiler uses the classification results in identifying the threads that may have unanswered questions. We evaluated the results with three human annotators, and 70-75% of the answers from the system were consistent with human answers.

In the next section we introduce the speech act categories that we have developed based on an analysis of student discussions. We then provide the details of the speech act classifiers including the steps for extracting features that are appropriate for classifying student discussions. The following section presents the thread profiler and shows how the SA classifiers are used in analyzing student interactions in discussion threads. Finally we present a summary and future work.

1. Modeling Threaded Discussions with Speech Acts

Table 1. Speech Act Categories for Individual Messages

Speech Act	Description	%
ACK-SUP-COMP	An acknowledgement, compliment or support in response to a previous message	8.5
INFORM	Information, Command or Announcement	6.7
ANS-SUG	A simple or complex answer to a previous question. Suggestion or advice	37.8
CORR-OBJ	A correction or objection (or complaint) to/on a previous message	9.7
ELAB	An elaboration (of a previous message) or description, including elaboration of a question or an answer	8.1
QUES	A question about a problem, including question about a previous message	29.2

Our discussion dataset is from an undergraduate Operating Systems course in Computer Science at the University of Southern California. The course is held every semester and the students have used the discussion board for the past five semesters. Our work focuses on the discussion corpus from the two most recent semesters. The corpus contains 475 threads with 1834 messages from 133 participants.

Unlike in a flat document set, in a threaded discussion each message plays a different role. For example, people may make arguments, support or object to points, or give suggestions. However, unlike prototypical collaborative argumentation where a limited number of members take part in the conversation with a strong focus on solving specific problems, online discussions have much looser conversational structure, possibly involving multiple anonymous discussants.

We have defined a set of speech acts (SAs) that relate pair of messages in discussion threads. A pair of messages may be labeled with more than one speech act. For example, the reply message could correct the original message as well as provide

an answer. A message can have SAs with respect to more than one message. Table 1 provides a definition of each speech act. We have explored different sets of speech acts based on assessment of human annotators and found that these categories are less confusing than other finer or coarser grained categories [5, 6]. In our corpus, questions (29.2%) and answers (37.8%) comprised the biggest portion of the corpus. This is consistent with the use of the discussion board as a technical question and answer platform for class projects. Figure 1 shows an example of a discussion thread with a sequence of question and answer exchanges.

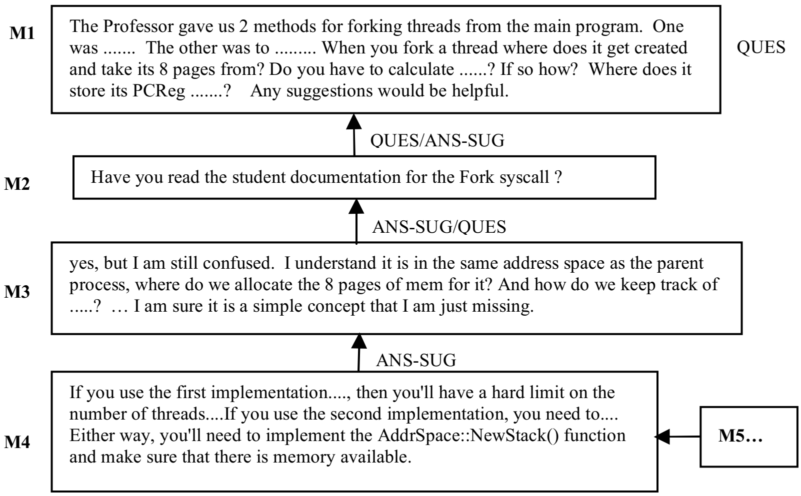


Figure 1: Example of a student discussion thread

2. Speech Act Classifiers: Identifying the Roles that Individual Messages Play

Our current work focuses on two SA classifiers: the question classifier (QC) and the answer classifier (AC). This section describes the details of the two classifiers.

2.1 Handling Incoherent and Noisy Discussion Data

We found that discussion data from undergraduate students are highly incoherent and noisy. The raw data includes humors and personal announcements as well as technical text. Student messages are very informal and there are high variances in the way they present similar information. A lot of messages on programming assignment include programming code. Besides typical data preprocessing and cleaning steps taken by many NLP systems, such as stemming and filtering, our system performs additional steps for removing noise and reducing variances.

We first remove the text from previous messages that is automatically inserted by the discussion board system when the user clicks on a *"Reply to"* button. We also apply a simple stemming algorithm that removes *"s"* and *"es"* for plurals. For discussions on programming assignment, the discussion included programming code fragments. Each section of programming code or code fragment is replaced with a single term called `code`. We then use a transformation algorithm that replaces

common words or word sequences with special category names. For example, many pronouns like *"I", "we"*and *"you"* are replaced by the symbol `categ_person` and sequences of numbers by `categ_number_seq`. For words like *"which", "where", "when", "who" and "how"*, we used the term `categ_wh`. Similar substitution patterns were used for a number of categories like filetype extensions (*".html", ".c", ".c++", ".doc"*), URL links and others. Students also tend to use informal words (eg: *"ya", "yeah", "yup"*) and typographical symbols such as smiley faces as acknowledgement, support or compliment. We transform such words into consistent words or symbols. We also substitute words like *'re, 'm, 've, don't,* etc. with *"are", "am", "have", "do not"*, etc. Finally, since speech act tends to rely more on surface word patterns rather than technical terms used, technical terms occurring in the messages were replaced by a single word `tech_term`.

2.2 Feature Selection

For our classifiers, we use N gram features that represent all possible sequences of N terms. That is, unigrams (single word features), bigrams (sequence of 2 words), trigrams (sequence of 3 words) and quadrograms (4 word sequences) are used for training and building the classifiers. There were around 5,000 unigrams or unique words occurring in the training corpus. Since the data was very noisy and incoherent, the feature space (where each feature could be a word or word sequence) is very large.

Table 2. Top N-grams based on Information Gain

Categ.	1-gram	2-grams	3-grams	4-grams
QC	? [categ_wh] will do confused	do [categ_person] [tech_term] ? can [categ_person] is there ? thanks	[categ_wh] should [categ_person] [categ_person] was wondering [or/and] do [categ_person] is there a [tech_term] ?	do [categ_person] have to do [categ_person] need to [tech_term] [tech_term] [tech..] ? is there a better does this mean that
AC	yes am helps but depends	look at [or/and] do seems like in [tech_term] stated in	look at the for example , . [categ_person] should let [me/him/her/us] know not seem to	[categ_person] am a [tech_term] do [categ_person] have to look at the [tech_term] in the same [tech_term]

We use Information Gain theory [7] for pruning the feature space and selecting features. Information gain value for a particular feature gives a measure of the information gained in classification prediction, i.e. how much the absence or the presence of the feature may affect the classification. First, we compute the information gain values for different N gram features extracted from the training data. For each feature, we compute two information gain values, one for QC and the other for AC. Subsequently, all the features (1-gram, 2-grams, 3-grams, and 4-grams) are sorted using the information gain. We use the top 200 features for each classifier. Some of the top N gram features for QC and AC are shown in Table 2.

2.3 SA Classifiers with SVM

The training set consisted of 1010 messages. A half of them were from one semester and the other half from the other semester. The test set had 824 messages. All the

threads were annotated by hand beforehand. The feature sets for training are extracted from the training corpus, as described above.

We use a linear SVM implementation [4] to construct SA classifiers. Linear SVM is an efficient machine learning technique used often in text classification and categorization problems. We constructed feature vectors for all the messages in the corpus. A feature vector of a message consisted of a list of values for individual features that represent whether the features existed in the message or not. We perform 5-fold cross validation experiments on the training data to set kernel parameter values for the linear SVM. After we ran SVM, the resulting classifiers (QC and AC) were then used to predict the SAs for the feature vectors in the test set. QC tells whether a particular message contains question content or not, and AC predicts whether a message contains answer content, i.e. answers or suggestions. The classification was then compared with the human annotations.

The resulting QC and AC had accuracies of 88% and 73% respectively. A similar pattern was observed among human annotators. The agreement ratio between the human annotators for questions (Agreement = 94.89%, kappa = 0.89) is higher than the one for answers (Agreement = 86.13%, kappa = 0.72) [8]. This could be because of higher degree of variances in the messages that contain answers or suggestions. Some of the answers and suggestions also take the form of questions (e.g. Message M2 in Figure 1), corrections, etc. making it difficult to classify the message.

3. Assessing When Interventions Are Needed: Thread Profiling with SA Classifiers

In order to assess discussion threads with SA classification results, we have developed a set of thread analysis rules. Our current analysis focuses on handling typical question answer patterns from students, as shown in Figure 2. That is, whether the given thread contains questions, and if it does, whether the questions were answered or not. The questions we formulated for thread assessment are the following:

- **(Q1) Was there at least one question in the thread? (Y/N)**
- **(Q2) Was the first question in the thread answered? (Y/N)**
- **(Q3) Were all the questions answered? (Y/N)**
- **(Q4) Were some of the questions answered? (Y/N)**

The thread analysis rules for answering these questions are shown in Table 3.

We randomly picked 55 threads from our test corpus and automatically classified each thread with our speech act classifiers and thread analysis rules. The threads that do not contain technical discussions such as humorous messages or personal information exchanges were not included. For each thread in the test corpus, we asked three humans to assess the thread and answer the above four assessment questions.

Some of the typical thread patterns that exist in the corpus are illustrated in Figure 2. The pattern (a) shows a thread that does not contain any question. Typically such a thread would contain only announcement messages. Many threads follow pattern (b) where the first message M1 is a question and subsequent messages M2 and M3 contain answers to this question. Pattern (c) has a Question-Answer-Question-Answer sequence, where all the questions in the thread are answered. Finally, the thread (d) illustrates the case of a *"hanging questions"*, where some of the questions are not answered. For example a student may try to understand the answer and ask a follow-up question (in M3) but does not receive any reply, leaving his/ her question unanswered.

Table 3. Thread Classification Rules

Rule	Description
1	- If QC = yes for any message in thread, answer "YES", else answer "NO" for question Q1.
2	- Find the first message m_i in the thread for which QC = yes - Find all messages m_j which are replies to m_i, - If AC = yes for any of the messages m_j, answer "YES" for question Q2, else answer "NO".
3	- Find the set of all messages M_q in the thread for which QC = yes - For every message m_i in the set M_q, find all the reply messages m_j - If all the messages m_i in M_q have at least one reply message m_j with AC=yes, then answer "YES" for question Q3, else answer "NO". - If some (at least one) of the messages m_i in M_q have a reply message m_j with AC=yes, then answer "YES" for question Q4, else answer "NO".

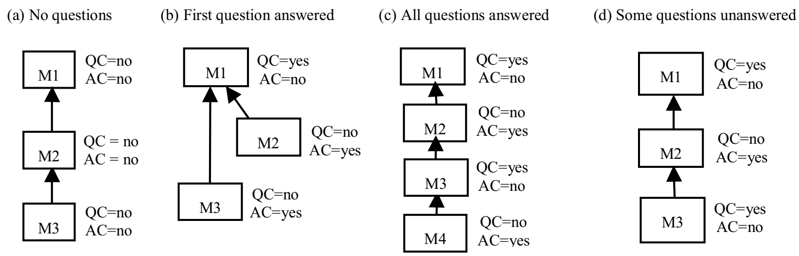


Figure 2: Example Question-Answer patterns

The answers from the system were compared with human answers. The results (percentage of agreement) from the comparison are shown in Table 4. For Q1, the average agreement among humans is about the same as the one between humans and the system. Since QC provides a high accuracy, it supports Q1 very well. For Q2--Q4, 70-75% of the answers generated from the system were consistent with human answers. We believe that a lower system performance in the latter case could be due to lower classification accuracy returned by the AC. For example, some messages contain answers in the form of questions, and some of the answers were presented in an indirect and informal manner spanning long sentences, which made the classification using word N-gram features harder.

Table 4. Thread Classification Agreement between humans and the auto-classifier (H = Human Annotators, M = automatic classifier system)

Assessment Question	Average agreement among humans	Average agreement between M and H
Q1	92.7%	92.7%
Q2	95.1%	74.5%
Q3	85.3%	70%
Q4	96.5%	75.8%

The results from this automatic thread profiling can potentially be used for ranking discussion threads. The threads with positive/Yes values for Q1 and negative/No values for Q2, Q3 and Q4 can be reported as the ones that may need instructor attention.

4. Related Work

There have been many attempts to assess collaborative activities. Various approaches of computer supported collaborative argumentation have been proposed. Machine

learning techniques have been applied to train software to recognize when students have trouble sharing knowledge in collaborative interactions [9].

Rhetorical Structure Theory [10] based discourse processing has attracted much attention with successful applications in sentence compression and summarization. Most of the current work focuses on sentence-level text organization [11] or the intermediate step [12]. Analyzing and utilizing discourse information at a higher level, e.g., at the paragraph level, still remains a challenge to the natural language community. In our work, we utilize the discourse information at a message level.

There has been prior work on dialogue act analysis and associated surface cue words [13, 14]. There have also been previous approaches like modeling Dialogue Acts for automatic tagging and recognition of conversational speech [15] and related work in corpus linguistics where machine learning techniques have been used to find conversational patterns in spoken transcripts of dialogue corpus [16]. Although they are closely related to our speech act analysis, it is hard to directly map the existing results to our analysis. The interactions in our corpus are driven by problems or questions initiated by students and often very incoherent. Carvalho and Cohen (2005) [17, 18] presented a dependency-network based collective classification method to classify email speech acts. Due to high variances and incoherence of student discussions, we face more challenges in data processing and feature selection.

There have been efforts to analyze interactions in on-line communities. For example, Talk-to-me [19] can predict the likelihood of a message receiving a reply based on the content of the message and the message sender. Our work provides complementary capability by providing SA-based interaction analysis capabilities. Also our analysis work is driven by requirements from instructors and students rather than need of general on-line communities seeking information.

Graph-based algorithms have been used in text mining, clustering, and other similar problems [20]. They could potentially be used to profile or find patterns in discussion threads, where threads could be represented by graphs and messages within a thread could represent nodes in the graph.

5. Summary and Future Work

This paper presents an approach for profiling student interactions in on-line collaborative discussions. Our work focuses on technical discussions among undergraduate students. In profiling discussion threads, we adopted the Speech Act theory and developed speech act classifiers using N-gram features and SVM algorithms. A set of thread analysis rules are used in assessing whether the given discussion thread has unanswered questions.

We found that automatic profiling of undergraduate student discussions is very challenging due to high incoherence and noise in the data. Especially messages that contain long sentences, informal statements with uncommon words, answers in form of question, are difficult to classify. We are looking at additional features such as features of neighboring messages and topic features [21] as candidates for improving the performance of the system. Our informal interview with human annotators indicates that features from previous messages may provide useful hints for the classifier. For example, if the previous message was a question, 84% of the reply messages present an answer [5]. We are currently exploring possible extensions to our classifiers based on

these ideas. We are also working on classifiers for the remaining SA categories such as objection or support so that we can analyze various types of student interactions.

Acknowledgement

This work is supported by the NSF CCLI-Phase 2 program (Award No. 0618859). We would like to thank Bernadette Aurelio and Cheng-Wei Lin for their data annotation work. We also thank Erin Shaw for her comments on an earlier draft.

References

[1] Feng, D., Shaw, E., Kim, J., and Hovy, E.H., 2006. An Intelligent Discussion-bot for answering student queries in threaded discussions. In Proceedings of Intelligent User Interface Conference, pp. 171-177
[2] Austin, J., 1962. How to do things with words. Cambridge, Massachusetts: Harvard Univ. Press.
[3] Searle, J., 1969. Speech Acts. Cambridge: Cambridge Univ. Press.
[4] Chang, C.-C. and C.-J. Lin, 2001. LIBSVM: a library for support vector machines.
[5] Kim, J., Chern, G., Feng, D., Shaw, E., and Hovy, E., 2006. Mining and Assessing Discussions on the Web through Speech Act Analysis, Proceedings of the ISWC'06 Workshop on Web Content Mining with Human Language Technologies.
[6] Kim, J. and Beal, C., 2006. Turning quantity into quality: Supporting automatic assessment of on-line discussion contributions, American Educational Research Association (AERA) Annual Meeting.
[7] Yang, Y., Pedersen, J.O., 1997. A Comparative Study on Feature Selection in Text Categorization, Proc. Of the 14th International Conference on Machine Learning, pp.412---420.
[8] Cohen, J., 1960. A coefficient of agreement for nominal scales, Educational and Psychological Measurement, 20, 37-46.
[9] Kolodner, J. L., & Nagel, K., 1999. The Design Discussion Area: A collaborative learning tool in support of learning from problem solving and design activities. Proceedings of CSCL'99, pp. 300-307.
[10] Mann, W.C. and Thompson, S.A., 1988. Rhetorical structure theory: towards a functional theory of text organization, Text: An Interdisciplinary Journal for the Study of Text, 8 (3), pp. 243-281.
[11] Soricut, R. and Marcu, D., 2003. Sentence level discourse parsing using syntactic and lexical information. In Proceedings of Human Language Technology conference - NAACL.
[12] Sporleder, C. and Lapata, M., 2005. Discourse chunking and its application to sentence compression. In Proceedings of Human Language Technology conference - EMNLP.
[13] Samuel, K., 2000. An Investigation of Dialogue Act Tagging using Transformation-Based Learning, PhD Thesis, University of Delaware.
[14] Hirschberg, J. and Litman, D., 1993 Empirical Studies on the Disambiguation of Cue Phrases, Computational Linguistics, 19 (3).
[15] Stolcke, A. , Coccaro, N. , Bates, R. , Taylor, P. , Van Ess-Dykema, C. , Ries, K., Shriberg, E. , Jurafsky, D. , Martin, R. , Meteer, M., 2000. Dialogue act modeling for automatic tagging and recognition of conversational speech. Computational Linguistics, v.26 n.3, p.339-373.
[16] Shawar, B. A.; Atwell, E. 2005. Using corpora in machine-learning chatbot systems. International Journal of Corpus Linguistics, vol. 10, pp. 489-516.
[17] Carvalho, V.R. and Cohen, W.W., 2005. On the collective classification of email speech acts. In Proceedings of SIGIR, pp. 345-352.
[18] Carvalho V., and Cohen, W., 2006. Improving Email Speech Act Analysis via N-gram Selection, HLT-NAACL, ACTS Workshop
[19] Arguello, J., Butler, B. S., Joyce, L., Kraut, R., Ling, K. S., & Wang, X., 2006. Talk to me: Foundations for successful individual-group interactions in online communities. In Proceedings of the ACM Conference on Human Factors in Computing Systems.
[20] Mollá, D., 2006. Learning of Graph-based Question Answering Rules. In Proceedings HLT/NAACL Workshop on Graph Algorithms for Natural Language Processing, 37-44.
[21] Feng, D., Kim, J., Shaw, E., and Hovy E., 2006, Towards Modeling Threaded Discussions through Ontology-based Analysis, In Proceedings of National Conference on Artificial Intelligence.
[22] Soller, A., and Lesgold, A., 2003. Computational Approach to Analyzing Online Knowledge Sharing Interaction, In Proceedings of Artificial Intelligence in Education.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

The Influence of Learner Characteristics on Task-Oriented Tutorial Dialogue

Kristy Elizabeth BOYER, Mladen A. VOUK and James C. LESTER
Department of Computer Science, North Carolina State University
Raleigh, NC 27695, USA
{keboyer, vouk, lester}@ncsu.edu

Abstract. Tutorial dialogue has been the subject of increasing attention in recent years, and it has become evident that empirical studies of human-human tutorial dialogue can contribute important insights to the design of computational models of dialogue. Students with particular characteristics may have specific dialogue profiles, and knowledge of such profiles could inform the design of tutorial dialogue systems whose strategies leverage the characteristics of the target population and address the communicative needs of those students. This paper reports on a study that was conducted to investigate the influence of learner characteristics (performance levels, self-efficacy, and gender) on the structure of task-oriented tutorial dialogue. A tutorial dialogue corpus was gathered from interactions transpiring in the course of problem-solving in a learning environment for introductory computer science. Analyses of the annotated dialogues suggest that the dialogue structure of (1) low-performing students differs significantly from that of high-performing students, (2) students with low self-efficacy differs significantly from that of students with high self-efficacy, and (3) males differs significantly from that of females.

Introduction

Providing intelligent tutoring systems with the ability to engage learners in rich natural language dialogue has been a goal of the AI & Education community since the inception of the field. With the investigation of tutorial dialogue in a number of systems devised to support a broad range of conversational phenomena (e.g., CIRCSIM [1], BEETLE [2], the GEOMETRY EXPLANATION TUTOR [3], WHY2/ATLAS [4], ITSPOKE [5], SCOT [6], ProPL [7] and AUTOTUTOR [8]), we have begun to the see the emergence of a core set of foundational requirements and functionalities for mixed-initiative natural language interaction. Moreover, recent years have witnessed the appearance of corpus studies empirically investigating speech acts in tutorial dialogue [9], dialogues' correlation with learning [10, 11, 12, 13], student uncertainty in dialogue [14, 15], and comparing text-based and spoken dialogue [5].

While all learners may engage in some universal form of tutorial dialogue, it may be the case that different populations of learners engage in qualitatively different forms of dialogue. It seems plausible that students with particular characteristics may have specific dialogue profiles, and knowledge of such profiles could inform the design of tutorial dialogue systems whose strategies leverage the characteristics of the target population and address the communicative needs of those students. This paper reports on a study

investigating the influence of learners' achievement levels, self-efficacy, and gender on task-oriented tutorial dialogue.

Given that human-human tutorial dialogue offers a promising model for effective communication [16], an experiment was conducted to study naturally occurring tutorial dialogues in a task-oriented learning environment. A text-based dialogue interface was incorporated into a learning environment for introductory computer science. In the environment, students undertook a programming task and conversed with human tutors while designing, implementing, and testing Java programs. To ensure that only natural language was used for communication and to eliminate the possibility of non-verbal communication such as gesture and body language, tutors were physically separated from students in an adjoining lab. The tutors' interface included a real-time synchronized view of the students' problem-solving workspace. Dialogues were logged and the resulting corpus was then manually annotated with tutorial dialogue acts. Analyses of the annotated dialogues suggest that the dialogue structure of low-performing students differs significantly from that of high-performing students, that the dialogue structure of students with low self-efficacy differs significantly from that of students with high self-efficacy, and that the dialogue structure of males differs significantly from that of females.

1. Task-Oriented Tutorial Dialogue Corpus and Dialogue Acts

While all tutorial dialogue is undertaken in support of learning tasks, one genre of tutorial dialogue is directly situated in the task at hand: these dialogues emerge as a result of the creation of learning artifacts such as designs, proofs, or computer programs. The domain investigated in this study, which has also been studied in the ProPL tutorial dialogue project [7], is that of computer programs. Here, students design, implement, and test programs (in the case of the study, Java programs) to meet a given specification. In the course of constructing the artifact, tutors and students pose questions to one another, tutors offer advice and feedback, and students make statements about the artifacts.

The Java Corpus was gathered by logging text-based dialogues between tutors and novice computer science students. The learning task was to complete a programming problem that required students to apply fundamental concepts such as iteration, modularization, and sequential-access data structures. Table 1 presents two sample annotated dialogue excerpts from the Java Corpus. In Dialogue Excerpt A, the tutor interacts with a low performing student, Student A, whose pre-test score was well below the median. The structure of Dialogue A illustrates many features commonly seen with low performing students, such as seeking to establish confirmation of a proposed plan before proceeding to implementation. In contrast to Dialogue A, Dialogue B illustrates some common characteristics of dialogues with high performing students. Student B seeks tutorial advice, then proceeds directly to implementation with no pre-emptive request for tutor feedback.

Table 1: Dialogue Excerpts

Dialogue Excerpt A	Dialogue Excerpt B
Tutor: Do you know how to do that? [TQ] **Student A:** Not really. [A] **Tutor:** Well we first need a new String that will hold zipCode's string value. [HA] **Student A:** So String z = zipCode? [RF] **Tutor:** Close. [PLF] **Tutor:** Then you can set that string equal to “”+zipcode. [HA] **Student A:** Ok so String z=“”+zipCode [RF] **Tutor:** Yeah. [PPF] **Student A:** Then what? [TQ]	**Student B:** So I need an if for each digit? [TQ] **Tutor:** One if should suffice, since it will be called in each iteration. [A] **Tutor:** You just need to know which element to reference. [A] **Tutor:** This would be done in the inner loop. [HA] **Student B:** Ok. [ACK] (*Student works for 2.5 minutes.*) **Tutor:** You've got the right idea. [UPF] **Student B:** Yeah, I had programmers block. [EX] (*Student works for 3 minutes.*) **Tutor:** Perfect. [UPF]

The Java Corpus consists of 5034 dialogue acts: 3075 tutor turns and 1959 student turns. The corpus was manually annotated with a set of tutorial dialogue acts designed to capture the salient characteristics of task-oriented tutorial dialogues. The coding scheme (Table 2) draws on a scheme devised for tutorial dialogue on qualitative physics problems [10]. While most of the acts in this scheme are present in the Java corpus as well, the particular dialogues in the Java corpus made it difficult to make judgements about short answer questions versus deep answer questions and to make fine-grained distinctions between hinting levels. The four-category scheme [9] and a more expansive non-tutorial dialogue act catalogue [17] also contributed common acts.

The entire corpus was annotated by a single annotator. In an agreement study to evaluate the consistency of the coding scheme and its application to the corpus, a second annotator labelled a subset of 969 acts (of the total 5034 acts in the corpus). This yielded a 0.75 Kappa between the two annotators, indicating a reasonable inter-rater reliability.

2. Experimental Design

Subjects were students enrolled in an introductory computer science course and were primarily freshman or sophomore engineering majors in disciplines such as mechanical, electrical, and computer engineering.

The corpus was gathered from tutor-student interactions between 35 students and 6 tutors during a one-week study. Tutors and students were completely blind to each other's characteristics as they worked together remotely from separate labs. Tutors observed student problem-solving actions (e.g., programming, scrolling, running programs) in real time. The tutors consisted of four graduate students and two undergraduates in computer science; five were male and one was female. Tutors all had some level of tutoring experience, and were not instructed about specific tutorial strategies.

Subjects first completed a pre-survey including items about self-efficacy, attitude toward computer science, and attitude toward collaboration. Subjects then completed a ten item pre-test over specific topic content. The tutorial session was controlled at 50 minutes for all subjects, after which subjects completed a post-survey and post-test containing variants of the items on the pre- versions. Any subject whose session was interrupted due to technical difficulties or external factors, or who completed the task early, was omitted from the data set for analysis ($n_{omitted}$=6).

Table 2: Dialogue Acts

	Act	Description	Examples
Student/Tutor	Task Question (TQ)	Questions about goals to pursue, ordering of goals, and the specific problem being solved.	“Where should we start?” “Should we use an array?”
Student/Tutor	Concept Question (CQ)	Questions about domain elements, concepts, or facts that would apply over many different problems.	“How do I declare an array?” “I don’t know how to write a loop.”
Student/Tutor	Answer (A)	Answers to a task or conceptual question.	“No.” or “Yes.” “We need to give it an index.”
Student/Tutor	Acknowledgement (ACK)	Positive acknowledgement of a previous statement.	“Okay.” or “Yeah.” “Alright.”
Student/Tutor	Extra Domain (EX)	A statement not related to the computer science discussion.	“Hello.” or “Sorry.” “Nice working with you.”
Student	Request Feedback (RF)	A request for evaluative feedback on completed or proposed problem solving steps.	“So should I do array[0] = 1?” “Does that look good?”
Student	Signal Non-Understanding (SNU)	An indication that a previous statement by the tutor is not clear.	“Kind of makes sense.” “Not really.” or “I’m confused.”
Student	Statement (S)	Assertion of fact.	“I am going to use a for loop.” “We need to initialize that variable.”
Tutor	(Un)Prompted Positive Feedback (UPF/PPF)	Positive feedback regarding problem solving action.	“Good job.” or “Looks great.” “Yep.”
Tutor	(Un)Prompted Lukewarm Feedback (ULF/PLF)	Partly positive, partly negative feedback regarding student problem solving action. .	“The first part is right, but…” “You’re close.” or “Well, almost.”
Tutor	(Un)Prompted Negative Feedback (UNF/PNF)	Negative feedback regarding student problem solving action.	“No.” “Actually, that won’t work.”
Tutor	Hint/Advice (HA)	Problem solving or conceptual hint or advice not in answer to a direct question.	“Each digit is represented by 5 bars.” “Let’s move on.”
Tutor	Request to Confirm Understanding (RCU)	A request for student to confirm or disconfirm understanding.	“Does that make sense?” “Are you with me?”

3. Results and Discussion

To compare dialogue structure based on learner characteristics, three partitioning criteria were applied to the student population: incoming performance level, self-efficacy rating, and gender. After briefly noting overall learning effectiveness, this section reports on dialogue structure characteristics for each student sub-population based on each of the three partitioning criteria.

For each student, learning gain was gauged by the difference between pre and post-test scores. On average, students scored 13% higher on the post-test than the pre-test. A pair-wise difference t-test indicates that the difference is statistically significant ($p < 0.05$).

3.1 Dialogue Profile Analyses

For each student dialogue session, the relative frequency of each dialogue act was computed as the ratio of the number of occurrences of that dialogue act to the total number of dialogue acts in the session. The relative frequency of dialogue acts was then computed for high-performing and low-performing students, for high-efficacy and low-efficacy students, and for female and male students. To determine whether intra-group differences in means were significant, t-tests were performed. Table 3 summarizes the relative frequency results, omitting all dialogue acts for which there was no significant difference by learner characteristics. Bolded values indicate statistically significant differences ($p \leq 0.05$). It should be noted that the three partitions are not independent. For example, high performing students were more often in the high self-efficacy group, and most females were in the low performing group. Despite these confounds, we draw meaningful conclusions by examining each learner characteristic individually.

Students were divided into low performing and high performing groups based on the median pre-test score. Analyses yielded the following findings: 1) High performing students made more acknowledgements, requested feedback less often, and made more declarative statements than low performing students. 2) Tutors paired with low performing students made more extra-domain statements, gave more prompted feedback, and made more requests for confirmation of understanding than tutors paired with high performing students.

Following an instrument devised by Bandura to measure domain-specific self-efficacy [18], students were asked to rate their confidence in being able to complete a programming assignment under a variety of circumstances. Because the problem used for this study was drawn from a standard problem set for the course, students had an experiential basis on which to judge their ability to complete the problem. Statistically significant differences in dialogue structure emerged when students were grouped by their confidence level regarding whether they could complete a simple laboratory assignment on their own. Analyses yielded the following findings: 1) Students in the high self-efficacy group made more declarative statements, or assertions, than students in the low self-efficacy group. 2) Tutors paired with low self-efficacy students gave more negative feedback and made fewer acknowledgements than tutors paired with low self-efficacy students.

Although females comprised a small number of our subjects, some statistically significant results emerged. 1) Women made more requests for feedback and fewer declarative statements than men. 2) Tutors paired with women gave more positive feedback and made more requests to confirm understanding than tutors paired with men.

3.2 Discussion

These findings extend those of previous studies investigating tutorial dialogue and learning effectiveness which have found correlations of dialogue structure and content with learning [11, 12, 13]. Of particular interest is a large spoken tutorial dialogue study conducted as part of the ITSPOKE project [10]. The ITSPOKE study found that student utterances exhibiting reasoning and reasoning-oriented questions posed by the tutor were

positively correlated with learning in a human-computer corpus, as were the introduction of new concepts in the dialogue by students in a human-human corpus. The Java Corpus study reported on here found that learner characteristics appear to significantly affect the structure of tutorial dialogue, and that both tutor and student dialogue acts appear to be affected by these differences. Tutors more often engaged in more extra-domain conversation, provided additional feedback, and more frequently engaged in discussions to gauge students' level of understanding when conversing with low performing, low efficacy, or female students. These same groups of students tended to request more feedback, make fewer declarative statements, and make fewer acknowledgements. It seems likely that learner characteristics affect (and are affected by) tutorial dialogue issues analogous to those bearing on help-seeking behaviors [19] and self-explanation [20].

These findings suggest that it may be possible to devise tutorial dialogue strategies that address the specific communicative needs of different groups of learners. Putting gender differences aside because of the limited data, several design implications could be considered for tutorial dialogue systems:

- **Encouraging Reflection:** If a student with low incoming performance initiates few requests for feedback, the system should consider taking remedial action such as asking task questions or concept questions to assess student understanding.
- **Giving Adequate Feedback:** Systems should be prepared to give prompted feedback more often when working with low-performing or low-efficacy students.

Table 3: Relative Frequency Results

	Dialogue Act	Pre-test Performance n_{high}=17, n_{low}=18		Self-efficacy Level n_{high}=19, n_{low}=16		Gender n_{female}=7, n_{male}=28	
Tutor and Student	Student Acknowledgement (ACK)	**High** **Low**	**10.3%** **6.3%**	High Low	9.1% 7.3%	Female Male	8.3% 8.2%
Tutor and Student	Tutor Acknowledgement (ACK)	High Low	2.3% 1.5%	**High** **Low**	**2.5%** **1.2%**	Female Male	1.2% 2.1%
Tutor and Student	Tutor Extra Domain (EX)	**High** **Low**	**6.9%** **9.4%**	High Low	8.5% 7.8%	Female Male	8.4% 8.2%
Student	Request Feedback (RF)	**High** **Low**	**1.2%** **2.2%**	High Low	1.6% 2.0%	**Female** **Male**	**2.8%** **1.5%**
Student	Statement (S)	**High** **Low**	**3.5%** **1.6%**	**High** **Low**	**3.5%** **1.4%**	**Female** **Male**	**1.2%** **2.8%**
Tutor	Prompted Positive Feedback (PPF)	**High** **Low**	**0.5%** **1.5%**	High Low	0.9% 1.3%	**Female** **Male**	**1.7%** **0.9%**
Tutor	Prompted Lukewarm Feedback (PLF)	**High** **Low**	**0.1%** **0.6%**	High Low	0.2% 0.5%	Female Male	0.6% 0.3%
Tutor	Prompted Negative Feedback (PNF)	**High** **Low**	**0.1%** **0.3%**	**High** **Low**	**0.0%** **0.4%**	Female Male	0.3% 0.1%
Tutor	Request Confirmation of Understanding (RCU)	**High** **Low**	**0.1%** **0.9%**	High Low	0.3% 0.8%	**Female** **Male**	**1.4%** **0.3%**

- **Making Acknowledgements:** When interacting with high-efficacy students, systems might give more acknowledgements than in their default setting; this may more accurately reflect the interaction expected when working with a human tutor.
- **Maintaining Conversational Comfort:** When interacting with students who have been deemed to be low-performing prior to the tutoring session, systems should consider making slightly more extra-domain statements, which could create a more conversational setting in which weaker students might feel more at ease.

4. Conclusion

Tutorial dialogue exhibits structural regularities that cut across learning tasks and domains. However, learner characteristics may profoundly affect the structure of tutor-student conversations. Analyses of task-oriented tutorial dialogues indicate that students' incoming performance levels, self-efficacy, and gender significantly influence the structure of dialogue. The findings suggest that learner characteristics may be considered in designing tutorial dialogue strategies that more effectively target the specific needs of students with particular characteristics.

The study reported here represents a first step toward understanding how learner characteristics affect the structure of tutorial dialogue. Several directions for future work appear promising. First, it will be important to explore the influence of learner characteristics on tutorial dialogue in the presence of surface level information about students' utterances. This line of investigation is of particular interest given recent results indicating that lexical cohesion in tutorial dialogue with low-performing students is found to be highly correlated with learning [21]. Second, a comparative analysis of alternate tutoring strategies on the effectiveness and efficiency of student learning for students with targeted characteristics will yield a clearer picture of the space of tutorial dialogue. Third, students' motivation and frustration undoubtedly influence (and are influenced by) the structure and content of tutorial dialogue, so developing a better understanding of these interrelationships will contribute to more effective tutorial dialogue management.

Acknowledgements

The authors wish to thank Tiffany Barnes and the members of the IntelliMedia Center for Intelligent Systems at NC State University for their insightful discussions on this research, along with the software development team of August Dwight and Taylor Fondren. Special thanks to Scott McQuiggan for assistance in framing the efficacy questions of this work and for assistance in preparing the manuscript. This work was supported in part by National Science Foundation Grants CNS-0540523 and REC-0632450. Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

[1] M. Evens and J. Michael, *One-on-One Tutoring by Humans and Computers.* Mahwah, New Jersey: Lawrence Erlbaum Associates, 2006.

[2] C. Zinn, J. D. Moore and M. G. Core, "A 3-tier planning architecture for managing tutorial dialogue," *Intelligent Tutoring Systems, Sixth International Conference (ITS 2002),* 2002.

[3] V. Aleven, K. Koedinger and O. Popescu, "A Tutorial Dialog System to Support Self-explanation: Evaluation and Open Questions," *Proceedings of the 11th International Conference on Artificial Intelligence in Education,* pp. 39-46, 2003.

[4] K. VanLehn, P. W. Jordan, C. P. Rose, D. Bhembe, M. Bottner, A. Gaydos, M. Makatchev, U. Pappuswamy, M. Ringenberg and A. Roque, S. Siler, R. Srivastava "The architecture of Why2-Atlas: A coach for qualitative physics essay writing," *Proceedings of the 6th International Conference on Intelligent Tutoring Systems,* pp. 158–167, 2002.

[5] D. J. Litman, C. P. Rosé, K. Forbes-Riley, K. VanLehn, D. Bhembe and S. Silliman, "Spoken Versus Typed Human and Computer Dialogue Tutoring," *International Journal of Artificial Intelligence in Education,* vol. 16, pp. 145-170, 2006.

[6] H. Pon-Barry, K. Schultz, E. O. Bratt, B. Clark and S. Peters, "Responding to Student Uncertainty in Spoken Tutorial Dialogue Systems," *International Journal of Artificial Intelligence in Education,* vol. 16, pp. 171-194, 2006.

[7] H. C. Lane and K. VanLehn, "Teaching the Tacit Knowledge of Programming to Novices with Natural Language Tutoring," *Computer Science Education,* vol. 15, pp. 183-201, 2005.

[8] A. Graesser, G. Jackson, E. Mathews, H. Mitchell, A. Olney, M. Ventura, P. Chipman, D. Franceschetti, X. Hu, M.M. Louwerse, N.K. Person, and the Tutoring Research Group, "Why/AutoTutor: A Test of Learning Gains from a Physics Tutor with Natural Language Dialog," *Proceedings of the Twenty-Fifth Annual Conference of the Cognitive Science Society,* pp. 474-479, 2003.

[9] J. Marineau, P. Wiemer-Hastings, D. Harter, B. Olde, P. Chipman, A. Karnavat, V. Pomeroy, S. Rajan, A. Graesser, and the Tutoring Research Group, "Classification of speech acts in tutorial dialog," in *Proceedings of the Workshop on Modelling Human Teaching Tactics and Strategies of ITS-2000,* pp. 65-71, 2000.

[10] K. Forbes-Riley, D. Litman, A. Huettner and A. Ward, "Dialogue-learning correlations in spoken dialogue tutoring," *Proceedings of the 12th International Conference on Artificial Intelligence in Education (AIED 2005),* pp. 225-232, 2005.

[11] M. G. Core, J. D. Moore and C. Zinn, "The role of initiative in tutorial dialogue," *Proceedings of the Tenth Conference on European Chapter of the Association for Computational Linguistics,* pp. 67-74, 2003.

[12] C. Rose, D. Bhembe, S. Siler, R. Srivastava and K. VanLehn, "The Role of Why Questions in Effective Human Tutoring," *Proceedings of Artificial Intelligence in Education,* 2003.

[13] S. Katz, D. Allbritton and J. Connelly, "Going Beyond the Problem Given: How Human Tutors Use Post-Solution Discussions to Support Transfer," *International Journal of Artificial Intelligence in Education,* vol. 13, pp. 79-116, 2003.

[14] J. Liscombe, J. Hirschberg and J. Venditti, "Detecting Certainness in Spoken Tutorial Dialogues," *Proceedings of Interspeech,* 2005.

[15] K. Forbes-Riley and D. Litman, "Using Bigrams to Identify Relationships Between Student Certainness States and Tutor Responses in a Spoken Dialogue Corpus," *Proceedings of the 6th SIGdial Workshop on Discourse and Dialogue,* 2005.

[16] M. T. H. Chi, S. A. Siler, H. Jeong, T. Yamauchi and R. G. Hausmann, "Learning from human tutoring," *Cognitive Science,* vol. 25, pp. 471-533, 2001.

[17] A. Stolcke, K. Ries, N. Coccaro, E. Shriberg, R. Bates, D. Jurafsky, P. Taylor, R. Martin, C. Van Ess-Dykema and M. Meteer, "Dialogue act modeling for automatic tagging and recognition of conversational speech," *Computational Linguistics,* vol. 26, pp. 339-373, 2000.

[18] A. Bandura, "Guide for constructing self-efficacy scales," in *Self-Efficacy Beliefs of Adolescents,* F. Pajares and T. Urdan, Eds. Greenwich, Connecticut: Information Age Publishing, 2006, pp. 307-337.

[19] V. Aleven, B. McLaren, I. Roll and K. Koedinger, "Toward tutoring help seeking: applying cognitive modeling to meta-cognitive skills," *Proceedings of the 7th International Conference on Intelligent Tutoring Systems,* pp. 227-239, 2004.

[20] M. T. H. Chi, N. Leeuw, M. H. Chiu and C. LaVancher, "Eliciting Self-Explanations Improves Understanding," *Cognitive Science,* vol. 18, pp. 439-477, 1994.

[21] A. Ward and D. Litman, "Cohesion and learning in a tutorial spoken dialog system," *Proceedings of the 19th International FLAIRS (Florida Artificial Intelligence Research Society) Conference,* 2006.

Collaboration

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Measuring the Effect of Collaboration in an Assessment Environment

Beatriz BARROS*, Ricardo CONEJO**, Eduardo GUZMAN**
** UNED, Spain,*
Juan del Rosal, 16, 28040 Madrid
*** Universidad de Málaga, Spain*
Boulevard Louis Pasteur, 35, 29071 Malaga

Abstract. This paper describes a computerized collaborative testing environment in which students can take an assessment session in groups. For each question, an individual response is firstly requested, then students are allowed to view the answers of the others and discuss the question. After that a final response is requested. This collaborative environment has been used to measure the effect of collaboration in test performance obtaining promising results that indicates that all students improve their performance no matter what their knowledge levels are.

Introduction

Collaborative learning scenarios should engage learners in cognitive and metacognitive activities to promote the conscious cooperative development of shared knowledge and to enrich their individual understanding of the world. Therefore, these scenarios should involve the learners in situations which require reflecting on their own knowledge as well as their colleagues' in a grounding process [2] in order to get, as result of this process, more refined and mature knowledge by themselves. There are several open challenges in this field, some of them related to this paper: first of all, the representation and quantitative measure of the process of collaboration; and second, the development of learning environments that focus the students in the learning objectives instead of dispersing the collaboration in aspects not related to the topic. The collaborative environment proposed in this paper defines semi-structured learning activities as collaborative scripts [9], which are not very rigid so they can have as rich, open and flexible a collaboration as possible.

Tests have been widely used for assessment. When they are carried out in groups, totally or partially, they are called collaborative testing [19]. Most of the research in this area has been done with paper and pencil tests. In this paper, an experiment with computerized tests in a collaborative scenario is presented. For each question in the test, students are allowed to give an individual answer that is shown to others; they can discuss and reflect on it, and subsequently, they have a second chance to submit an individual answer. For students, it is motivating [10] to know the answers of their colleagues and to chat about the answers during an assessment session. In this context, it is to be expected that the students are going to centre the discussion on the question meaning, on understanding their failures as well as their colleagues' and, in the case of conflict, to argue which of the answers is the right one. This scenario is interesting for studying the collaboration the partners discuss about the content to be learned and do not scatter the conversation. Berry[3]observed that working in small groups on assessment activities gives students an opportunity to speak about the topic and thereby sharpen their skills and understanding.

The collaborative assessment environment also incorporates a structured communication module, a chat tool that allows having argumentative discussions with labelled utterances in groups [17]. The system registers all the collaborative actions into a log file for off-line analysis. This environment is especially attractive for research because: a) it is designed as a learning activity that combines individual tasks with collaborative tasks b) it is a scenario that facilitates students' reflection and discussion on its own knowledge and the others' in a situated learning environment [11]; c) it forces students to adopt socially-acceptable ways of resolving conflicts [18]; d) students can build upon each other's knowledge with positive performance outcomes [8]; e) it introduces collaboration in scenarios that commonly are meant for individual learning; and f) it is a first step to study empirically how collaboration has influence on the performance of the learner, because (s)he is evaluated before and after the collaboration process.

In section 1, some points related with this work are going to be discussed. Next, the research scope and motivation of the paper are presented. In section 3 the collaborative environment is described and then, the experiments done with groups of students and the most significant results obtained from the evaluation process. The paper finishes with some conclusions and future proposals for working.

1. Related work

Some work has implemented and played on experiences combining computerized tests and collaboration, some of them in the classroom. Some studies [5] [13], observed that retentions and results of course content increases with the use of collaborative testing. Furthermore, Simkim [19] performed two experiments in the classroom with multiple-choice questions and obtained higher group scores, compared with individual scores. He also found that "*group exams encourage faculty to ask more challenging questions than they might otherwise and potentially increasing the amount of learning in the classroom*". [4] argue that there is no effect of collaborative testing performance in questions about theories but students should gain from collaborative testing when questions are about concepts.
The above-mentioned experiences have been done by means of paper and pencil tests, posing a whole set of questions to each student and later, a similar set in the collaborative mode. As far as we know, there are no published data about existing computerized collaborative assessment tests that interleaves individual and collaborative responses. Two interesting experiments [12] have been carried out in a web-based environment that combine argument-based collaboration with questionnaires (about opinions that are not correct or incorrect answers) and sharing facilities to work and reflect in groups. It uses a collaborative script, called ArguedGraph, with five phases that combines individual, collective and collaborative phases, all of them supported by a computer. During group phases, students have a chat tool to discuss and write their arguments. Similarly, they use a collaborative script that combines individual an collaborative phases (with a chat tool), but with different purposes. There has been also some research on peer assessment where students evaluate the answers of others. Our research is different from all these, because: (1) questions are used for assessment; (2) our script interleaves individual and collaborative phases, (3) the answers are finally evaluated by the computer, and (4) the objective of our experiments is measure the effect of collaboration in test performance.

It has been reported that on-line interactions among students make positive contributions to students' learning [14]. Furthermore, [7] observed that the analysis of students'

contributions to online discussions provides evidence of effective collaboration in an interactive e-learning environment. A text-based online conversation tool that allows performing on-line interactions could be implemented using a chat [15]. Chat tools are being used in several collaborative learning environments, as a free conversation tool [21], or with sentences openers [20], or with typed messages, that represent conversation acts with [1] [19] or with explicit reference to a shared document [16]

2. Research scope and motivation

An experiment for a computerized collaborative test has been designed, implemented and evaluated with real users. The first step was to design a collaborative script (Figure 1). The activity is designed for small groups of (2-5) students that take a test of *n* questions. The answer to each question is organized in three phases. Students interact synchronously and the conditions to pass to the next phase are to complete the previous one. The first phase consists of answering a question on their own. During the second phase, students share the results with their colleagues and discuss the answers given in the previous phase. In the third phase, each student answers the same question again individually. It is not compulsory to get to an agreement for this second answer. Each student can decide to keep or modify his/her first answer. This script is repeated for all questions in the test.

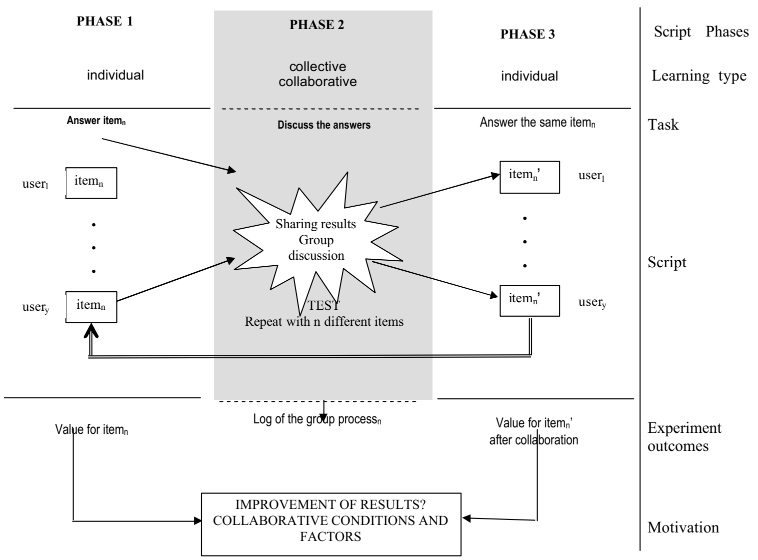


Figure 1. *Collaborative Script for the collaborative computerized tests, outcomes and motivation*

The objective of the experiment is to explore if students increase their performance when they work in groups and to study if the collaborative discussion process is related with this improvement.

3. The assessment environment

The script has been implemented in a web-based collaborative environment that manages the script sequencing, activates the collaborative interface and invokes the assessment tool SIETTE [6]. The assessment tool collects the answers of the students, and the events that

describe the activity of the users during the collaboration phase, that is, if they have seen the answers of others, the chat messages exchanged, etc.

SIETTE is a Web-based system for building and administering computerized tests. In fact, SIETTE is a suite of tools which implements all the stages of test construction, delivery and result analysis.

The empirical studies presented in this paper have been carried out using an extended version of SIETTE supporting student collaborative testing. We have built an envelope around SIETTE which provides all the mechanisms needed to allow and synchronize the collaboration among students. Briefly, this envelopment, that we have called *collaborative frame*, has been implemented by means of a Java applet shown as a plug-in in the left side of the web browser window. This frame is only shown in the SIETTE's *virtual classroom* when the student is taking a collaborative test, and it is in charge of controlling all the aspects related to the student collaboration. Because of the use of synchronous communication, many awareness facilities have been included in the interface so the students can know where their colleagues are. The *collaborative frame* submits and retrieves information from a server, implemented by means of a Java servlet. Such server implements a HTTP-based communication protocol between SIETTE and the collaborative frame of each student. Moreover, this server traces all the actions accomplished by the students as well as the messages they interchange. All tests available in SIETTE can be posed in the collaborative mode. The system with or without the collaborative support is available at http://www.lcc.uma.es/siette.

Figure 2 shows the environment while two students are taking a collaborative test. The right frame, the assessment-frame (labelled with ①), is used by SIETTE to pose questions, and to show the answers given by other students. The left frame is composed by the awareness-frame (upper, labelled with ②) and the communication-frame (below, labelled with ③). The awareness-frame depicts the evolution of the students involved in the test (including himself). The information of each student is shown in a different row. The first row always corresponds to the current student. Each row begins with the student nickname, followed by the order of the question he/she is answering at this moment. Next to it, a set of status bars displays information about which stage the student is at, i.e. *individual response*, *discussion*, *group response* and *finished*. Such bars have different colours depending on which question the student is at, regarding the current student. Accordingly, the current student bar is always shown in green. All those students answering the same question are shown in green, those answering a former question are in red, whereas those in posterior questions in blue. Finally, each row has a button which allows the student to query the answers selected by the others. This button is only enabled during the collaborative stages of *discussion* and *group response*.

The communication-frame has a chat window. Its upper part is the post panel. It is formed by a hierarchy of messages submitted by all the students involved in the test. As it can be seen in the lower part, students can send four types of message, i.e. comments, justifications, questions and answers. The chat is only enabled during the *discussion* and *group response* stages of each question. In addition, every time the chat is disabled, the post panel contents are cleared and its messages are not kept between two questions. That is, when the chat is enabled, the post panel does not contain the messages sent in former questions. Furthermore, a student can only see those messages sent during the question which he/she is currently contemplating. This means that if two students (A and B) are

trying different questions (Q_A and Q_B respectively), the one who is in the former question (for instance A), will only be able to read the messages sent by the other (student B) when student A arrives at question Q_B.

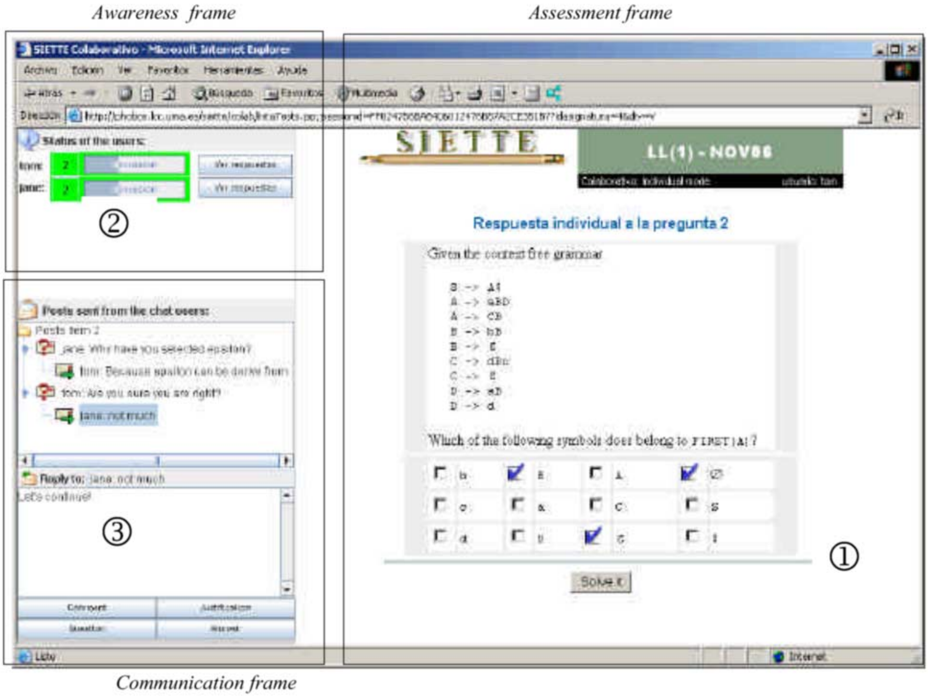


Figure 2: *Interface of the collaborative assessment environment*

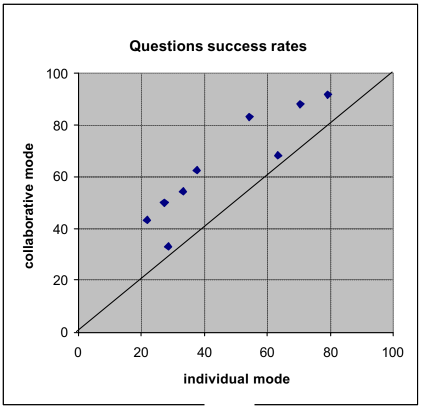
4. Experiments

The evaluation was carried out by using a test that contains 10 questions about Compiler Construction. In particular, we focus on the LL(1) techniques and asked questions about finding the elements of the set obtained by FIRST() and FOLLOW() functions of a certain context free grammar. However, the system is completely domain independent. All questions were multiple choice so that the students were asked to select the appropriate set of symbols among 16 choices. That means that the probability of finding the right answer just by guessing is $1/2^{16}$, that is, almost 0.

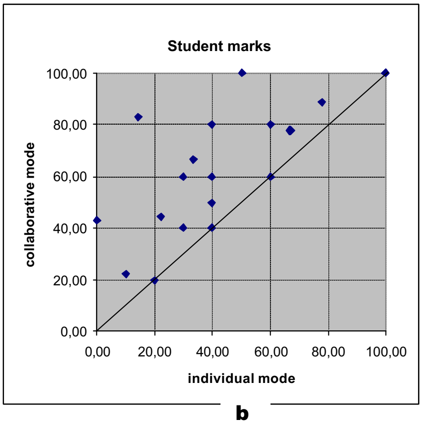
Among these questions, the first one was easier than the others, and was included just for training the student to use the collaborative environment. This question has been removed in the results analysis, so the test can be considerd having just 9 questions. The evaluation involved a total of 24 students, in two different sessions of 7 and 17 students each. Students were randomly divided in 9 groups of 2 people and two groups of 3 people, one for each session.

First of all, we analyze the average of correct responses that has been given before the discussion phase (individual mode), and after (collaborative mode). Figure 3a shows the results by each question. As it is shown a better performance is obtained for all questions.

In the first case the average success rate was 45,8% and in the second 63,7%, The difference is statistically significant (p < 0,05), even more so if we consider that the answers were given one after the other and that the probability of guessing is almost 0.



a



b

Figure 3: *Questions and students performance in individual and collaborative modes*

A reciprocal study has been done by considering the students' marks, obtaining similar results. As it was expected, figure 3b, all of the students improved their marks after the collaboration phase.

To deeply analyze what is going on inside the groups, we have divided the students in two classes according to their own results in the individual mode: the B-class which is formed by those students below the average, and the A-class, formed by those above the average. The data obtained indicates that all groups improve their mark in the collaborative mode. (table 1). The second column in this table represents the average of the differences in the marks (percentage of correct answers) obtained before and after the collaboration. It is not surprising that the B-class improve more than the A-class, simply because they can improve more. A new measure named "*relative improvement*" x is proposed to compare the results of the two classes, which is given by the formula:

$$x = \frac{M_C - M_I}{100 - M_I}$$

Where M_I represents the mark (percentage of correct answers) obtained before the collaboration, and M_C represents the mark (percentage of correct answers) after the collaboration phase. This result provides evidence that the relative improvement is similar in both classes.

Table 1: *Performance of students according to their estimated knowledge level*

	Number of cases n	Average of absolute improvement	Average of relative improvement $\bar{x}$	Standard deviation of the relative improvement s	Confidence interval at 95% $\bar{x} \pm t_{0.025} \frac{s}{\sqrt{n}}$
A-class	10	+11.7%	0,30	0,32	0,30±0,23
B-class	14	+23.1%	0,31	0,25	0,31±0,15

We can also analyze the results according to the group companion, and divide the sample in two classes: Those students that have a companion with higher individual marks (noted as L-class) and those that do not have a companion with higher individual marks (H-class). That is, in each group the student with the higher individual score is considering to be of H-class and the rest are L-class. Because there were 2 groups of 3 people the L-class and H-class have different sizes. 2 cases were discarded because they obtained the same mark as their companion.

Table 2: *Performance of students according to their relative knowledge level in their groups*

	Number of cases n	Average of absolute improvement	Average of relative improvement $\overline{x}$	Standard deviation of the relative improvement s	Confidence interval at 95% $\overline{x} \pm t_{0.025} \frac{s}{\sqrt{n}}$
H-class	10	+6,33%	0,16	0,19	0,16±0,14
L-class	12	+30,34%	0,47	0,27	0,47±0,17

The results (table 2) indicate that both classes improve their results ($p<0.05$), and that there is a significant difference ($p<0.05$) between the L-class and H-class. That is, it can be statistically concluded that those students that have at least one companion with a higher knowledge level, improve more than those that have the highest knowledge level in their groups.

The last experiment concerns the collaboration within groups and its relation to student improvements, measured as the average of the "*relative improvement*" of the student in the group as it was defined before. We have defined four indicators of the collaboration inside a group: a) The total number of messages in the chat room, b) The total length of all messages, c) The average length of messages, and d) the number of interactions, that is the number of response messages to previous messages posed by others. The last indicator tries to distinguish between those messages posted just by the same student, (a stand alone behaviour), from those that are posted as part of a conversation. We have obtained the correlation coefficient for each indicator, and found that there is a positive correlation between the average relative improvement and the total number of messages in the chat room ($r=0.74$) and with the interaction indicator ($r=0.71$). In fact both indicators were highly related in this case. These results are statistically significant with $p<0.05$. Another good indicator is that in 70% of the cases, the students have changed their previous responses after the collaboration phase, and in 66% they have agreed on a common answer.

5. Discussion and Conclusions

From our experiments it can be statistically concluded that collaboration, as has been defined in our web-based assessment environment, improves assessment performance for all students. Even those students that are the best in their group improve their performance as a consequence of the dialogue with other students, probably because they have to reflect on their answers to explain them to others.

In the experiment presented in this paper, the results before and after collaboration have been observed as well as the collaboration process in terms of three single indicators, number of messages, size of the messages and interaction level.

These data have been contrasted in relationship to the performance of the students. Results show that the bigger the interaction level within the group the better the student's performance. Furthermore, an interesting property of the collaborative assessment environment presented in this paper is the possibility of a quantitative measurement of the effects of collaboration that allows a deeper analysis of what is going on inside the groups.

References

[1] Barros, B.. & M.F.Verdejo (2000)."Analysing student´s interaction process for improving collaboration: the DEGREE approach", *IJAIED* 11, pp. 221-241

[2] Baker, M.J., Hansen, T., Joiner, R. & Traum, D. (1999). The role of grounding in collaborative learning tasks.In P. Dillenbourg (Ed.), *Collaborative Learning : Cognitive and Computational Approaches*, pp. 31-63. Amsterdam : Pergamon / Elsevier Science.

[3] Berry, J. and Nyman, M.A. (2002), "Small-Group Assessment Methods in Mathematics" *Int. Journal of Mathematical Education in Science and Technology* vol. 33, no. 5, pp. 641-64 9

[4] Breedlove, W., Burkett, T. & Winfield, I. (2004) Collaborative Testing and Test Performance, *Academic Exchange Quarterly*, Fall 2004, Vol. 8, Is.3.

[5] Cortright,R.N., Collins,H.L., Rodenbaugh, D.W. & DiCarlo, S.E. (2002) "Student Retention of Course Content is Improved by Collaborative-Group Testing", *Advances in Physiololgy Education*, 26, pp. 37-41.

[6]. Conejo, R., Guzmán, E. Millán, E. Trella, M. , JPérez-de-la-Cruz J.L. & Ríos A. (2004) SIETTE: A Web-Based Tool for Adaptive Testing. *IJAIED*, 14, pp. 29 61.

[7] Curtis, D.D. & Lawson, M.J. (2001) "Exploring Collaborative Online Learning", *JALN,* Vol. 5, Iss. 1, February 2001, pp. 21-33.

[8] Damon W. & Phelps, E. (1989) Critical distinctions among three methods of peer education; *International Journal of Education Research,* 13, PP. 9-19.

[9] Dillenbourg, P. (2002). Over-scripting CSCL: The risks of blending collaborative learning with instructional design. In P. A. Kirschner *Three worlds of CSCL*, pp. 61-91.

[10] Graham, R.A. & Graham, B.L. (1997) "Cooperative Learning: The benefits of Participatory Examinations in Principles of Marketing Classes", *Journal of Education Business*, Vol. 72, no. 3, January/February, pp1. 149-152.

[11] Henning, P. H. (2004) Everyday Cognition and Situated Learning, In D. H. Jonassen (Ed.), *Handbook for Research in Educational Communications and Technology*, 2nd Edition. (pp. 143-168). Mahwah, NJ: Lawrence Erlbaum Associates.

[12] Jermann, P.,& Dillenbourg, P. (2002) "Elaborating new arguments through a CSCL Script" , Andriessen, J., Baker, M., Suthers, D. (eds.), Arguing to Learn: Confronting Cognitions in Computer-Supported Collaborative Learning environments, pp. 1- 6.

[13] Hite, P. A. (1996). "An experimental study of the effectiveness of group exams in an individual income tax class". *Issues in Accounting Education*, 11(1), 61-75.

[14] Laurillard, D. M. (1999). A conversational framework for individual learning applied to the 'learning organization' and the 'learning society.' *Systems Research and Behavioural Science, 16*(2), 113–122.

[15] Mazur, J. M. (2004). Conversation analysis for educational technologists: Theoretical and methodological issues for researching the structures, processes and meaning of on-line talk. In D. H. Jonassen (Ed.), *Handbook for Research in Educational Communications and Technology*, 2nd Edition. (pp. 1073-1098). Mahwah, NJ: Lawrence Erlbaum Associates.

[16] Mühlpfordt, M. & Wessner, M. "Explicit referencing in chat supports collaborative learning" *CSCL'2005*

[17] Pask, G. (1976). *Conversation theory: Applications in education and epistemology.* Amsterdam: Elsevier.

[18] Simkin. M.G. (2005) An Experimental Studyof the Effectiveness of Collaborative Testing in an Entry-Level Computer Programming Class, *Journal of Information Systems Education*, Fall 2005, Vol. 16, pp. 273-280.

[19] Sheremetov, L., Guzman-Arenas, A. (2002) "EVA: an interactive Web-based collaborative learning environment", *Computers & Education,* 39, 161–182

[20] Soller, A. (2001) Supporting Social Interaction in an Intelligent Collaborative Learning System. *IJAED*, 12, 40-62.

[21] Veerman, A.L., E.B. Andriessen & Kanselaar, G. (1999) Collaborative Learning through Computer-Mediated Argumentation, *CSCL'99*

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Tutorial Dialogue as Adaptive Collaborative Learning Support

Rohit KUMAR, Carolyn P. ROSÉ, Yi-Chia WANG, Mahesh JOSHI and Allen ROBINSON
Carnegie Mellon University, 5000 Forbes Ave., Pittsburgh, PA 15213, USA
{rohitk,cp3a,yichiaw,maheshj,alr}@cs.cmu.edu

Abstract. In this paper we investigate the role of reflection in simulation based learning by manipulating two independent factors that each separately lead to significant learning effects, namely whether students worked alone or in pairs, and what type of support students were provided with. Our finding is that in our simulation based learning task, students learned significantly more when they worked in pairs than when they worked alone. Furthermore, dynamic support implemented with tutorial dialogue agents lead to significantly more learning than no support, while static support was not statistically distinguishable from either of the other two conditions. The largest effect size in comparison with the control condition of individuals working alone with no support was Pairs+Dynamic support, with an effect size of 1.24 standard deviations.

Keywords. Collaborative Learning, Dynamic Support, Tutorial Dialogue Agents

Introduction

In order to encourage productive patterns of collaborative discourse, researchers both in the Computer Supported Collaborative Learning (CSCL) tradition and the Educational Psychology tradition have separately developed approaches to using scripts for scaffolding the interactions between students, to help coordinate their communication, and to encourage deep thinking and reflection [1]. Previous approaches to scripting have been static, one-size-fits-all approaches. In other words, the approaches were not responsive to what was happening in the collaboration. This non-adaptive approach can lead to over scripting [2] or interference between different types of scripts [3]. With these things in mind, and considering that ideally we would like students to internalize the principles encoded in the script based support, a more dynamic and potentially more desirable approach would be to trigger support based on observed need and to fade scaffolding over time as students acquire the skills needed to collaborate productively in a learning context. The concept of adaptive collaborative learning support has already been evaluated in a Wizard-of-Oz setup and found to be effective for supporting learning [4]. As a proof of concept of the feasibility and potential impact of such an approach, in this paper we describe an evaluation of a simple but fully-automatic approach to adaptive collaborative learning support.

Context sensitive or need based support necessitates on-line monitoring of collaborative learning processes. And, indeed, there has been much work in the computer supported collaborative learning community on modeling the process of collaborative learning with coding schemes applied to corpus data by hand. Our own recent work towards automating the application of a well established collaborative learning process analysis coding scheme [5] demonstrates that patterns that indicate trouble in a collaborative

discourse can be detected with a high degree of reliability and thus that a more dynamic support approach for collaborative learning is feasible [6]. The technology for automating such analyses is now publicly available[1]. So the time is ripe for exploring the use of this technology for triggering adaptive collaborative learning support.

In this paper we introduce and evaluate a new adaptive support mechanism specifically designed to draw out reflection using conversational agents that engage students in directed lines of reasoning [8]. This new collaborative learning support approach makes use of tutorial dialogue technology in the context of a simulation based learning task for college level thermodynamics instruction, building on previous results demonstrating the effectiveness of these tutorial dialogue agents for supporting learning of individuals working alone in this domain [9].

1. Architecture for Adaptive Collaborative Learning Support

Figure 1 displays our general architecture for triggering adaptive collaborative learning support, which was used in the work reported in this paper. This architecture is meant to be general purpose. Here we describe the basic architecture and the three main ways it can be tailored to different learning contexts.

Each student has their own chat client, as displayed at the bottom of the diagram. Each chat client connects to a central Server/Coordinator that maintains a dialogue history with the contributions of all of the group members, including the contributions of the support agent. Thus, each student's client can display the entire interaction. As student contributions are accumulated in the history maintained by the Sever/Coordinator, these contributions are also passed through a Filter that is tailored to identify a pre-selected set of conversational events. As these events are detected, notification of these events is passed to the Interaction Model, which maintains the conversational state. Certain states trigger support from the Chat Agent.

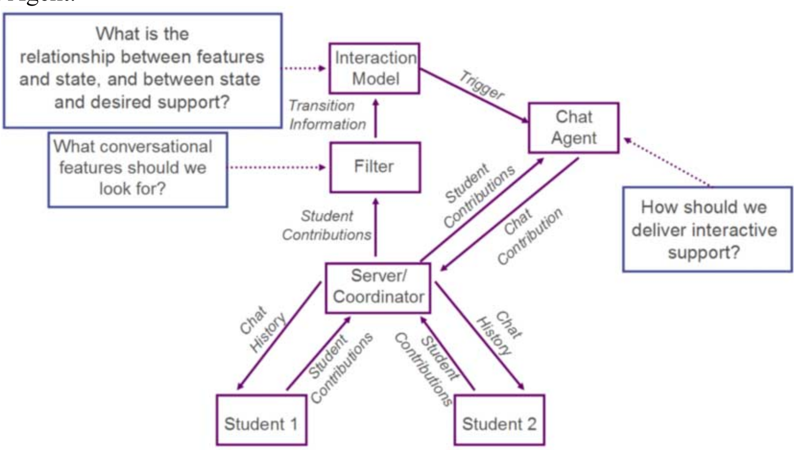


Figure 1 This figure displays our general architecture for enabling adaptive collaborative learning support. It can be customized in three ways: (1) the set of conversational features that the Filter is set up to identify in running conversations, (2) the way detected features are accumulated into a dialogue state as well as which states trigger support, and (3) the form of the support that gets triggered.

The first way this architecture can be tailored is through the Filter. In the work reported in this paper, we use a simple topic oriented filter, which we describe in the next section. However, we could have taken a an approach where we detected events associated with

[1] TagHelper tools can be downloaded from http://www.cs.cmu.edu/~cprose/TagHelper.html.

ways in which the students are interacting with one another, as in our previous work on automatic collaborative learning process analysis [6]. The second means by which this architecture can be tailored is in the Interaction Module. In the current work, this module is implemented in a trivial manner. Each time a topic is detected that has not been previously detected within the conversation, it triggers support for reflection related to that topic. However, we could have taken an approach whereby we looked at patterns of argumentation behavior that are indicative of unproductive collaboration. The final manner in which this architecture can be tailored is the form of support offered to students by the Chat Agent. In our previous work we have evaluated support in the form of prompts requesting further elaboration on explanations [4] or prompts suggesting new categories of ideas to consider [13]. In this study, support is offered in the form of directed lines of reasoning.

2. Experimental Infrastructure: The CycleTalk Chat Environment

We are conducting our research in the domain of thermodynamics, using as a foundation the CyclePad articulate simulator [7]. CyclePad was developed with the intention of allowing students to engage in design activities earlier in their engineering education than was possible previously. Our explorations of CyclePad use focus on design and optimization of thermodynamic cycles, specifically Rankine cycles.

2.1 The CycleTalk Chat Environment

In our previous work, we have integrated the CyclePad simulation environment with example tracing tutoring systems [10] using the Cognitive Tutor Authoring Tools [11]. We have evaluated this integrated version in our previous studies [9,12]. The integration with example tracing tutor technology allows us to offer students the opportunity to ask for hints as they work with the environment. In our previous studies [4], we have demonstrated that integrating tutorial dialogue agents with this environment significantly improves student learning. The integration of CyclePad with hints provided by example tracing tutors and tutorial dialogues provided by the TuTalk tutorial dialogue engine [8] comprises what we refer to as the original CycleTalk system. In our previous work, we have demonstrated that students learn significantly more from CycleTalk when dialogues are provided in addition to hints rather than only hints.

In the current study, students complete all of the instruction we used in our previous studies before they even take the pretest. However, the tutorial dialogue agents that were integrated with the CycleTalk environment previously are now used as part of the collaborative learning support we want to test as an intervention. Thus, although we still use the CycleTalk system during the instruction that occurs before the pretest, we use a version of the system without the tutorial dialogue component while students are working through those materials.

In the previous section we described the role of the Filter in our architecture for adaptive collaborative learning support. In the CycleTalk Chat Environment, the Filter was constructed semi-automatically from a topic segmented and labeled corpus collected in a previous CycleTalk study involving human tutors [12]. That corpus is comprised of 379 topic segments belonging to 16 topics. The first step of our process was to identify the terms that were strongly associated with each of these topics. We did this using a standard metric used in the information retrieval community referred to as term frequency inverse document frequency or TF-IDF. Using this metric, words that occur frequently within a document but relatively infrequently across documents receive a high weight for that document, indicating that term is very informative about what is special about that document. The result of this

process was a weight associated for each phrase per topic. Using these weights, a topic can be selected by averaging the term weights for all unigrams and bigrams identified in a student contribution for each topic, and selecting the topic that receives the highest weight. Based on an error analysis of the performance of this filter, we determined that the biggest source of errors was on false positives, or identifying a spurious topic where it was not raised. Thus, we implemented an additional filter that attempts to distinguish important contributions where a topic is raised from those where no topic is raised. We used TagHelper tools [14] to construct a support vector machine (SVM) model based on our topic labeling. Thus, the final topic filter first uses this model to determine whether a topic may have been raised, and then used the filter described above to determine which topic it is most likely to be.

The Filter just described identifies when concepts are raised that are related to one of the 7 directed lines of reasoning related to the concepts we want to stimulate reflection about. Whenever a concept is identified in the student input, a trigger is sent to the Chat Agent to launch the corresponding knowledge construction dialogue. Beyond this, the Interaction Model also includes a collaboration policy that aims to maintain the students' engagement and to motivate the students to participate equally in the conversation. First, the student(s) are prompted once every minute with a hint about a concept that has not been discussed in the conversation so far. The prompts are addressed to a specific student. In case multiple students are involved in the conversation, the student contributing fewer concepts is more likely to be prompted. Students are also prompted with a generic prompt at the end of every knowledge construction dialogue, to indicate the end of a guided interaction and to encourage the students to get back into idea generation and information gathering.

The Chat Agent delivers two types of support. The first form of support is the hints triggered by the collaboration policy described above. The form of the hints was based on a design tested in previous collaborative idea generation studies [13]. These prompts are meant to stimulate thinking towards an idea without giving the idea away. One example meant to elicit the idea of manipulating the maximum pressure of the cycle is “Try to think of an idea related to manipulating a property of the pump.”

The second form of support is the directed lines of reasoning implemented with the TuTalk dialogue engine. We use a subset of the same dialogue agents that were developed and evaluated in our previous work [4]. In the context of the current study, the purpose of these directed lines of reasoning is to support the student in reflecting on how concepts from the instructional phase can be applied in the exploratory phase. Each of 7 directed lines of reasoning focus on one or two key relationships between cycle variables, for example how the maximum temperature of the cycle affects the thermal efficiency of the cycle. These directed lines of reasoning lead the student to reflect on the relationships between the variables and what implications that relationship should have on design and optimization of thermodynamic cycles.

Here is an example interaction involving two students and the CycleTalk agent:

ST1: well what valtues do u have for the reheat cycle ?
Tutor: Let's think about the motivation for Reheat. What process does the steam undergo in the Turbines ?
ST2: for some reason I said temperature at turbine to be like 400 C
Tutor: Steam undergoes Isentropic Expansion in a turbines that converts the heat energy of the steam into work by rotating the blades. What happens to the Quality of steam during Expansion ?
ST2: shuoldnt quality stay the same through the turbine
Tutor: As steam expands in a turbine, its quality degrades as the moisture content in the steam increases. Should we use steam of degraded quality in the turbines ?

ST1: ya
Tutor: We should NOT use low quality steam in turbines as it condenses on the turbine blades and damages the turbine running at high speeds Does this constrain the amount of work steam can do in turbines ?
ST1: oh. no.
Tutor: Since quality of steam falls gradually in the turbines, we design turbines such that the quality remains above acceptable levels [0.85]. …

3. Method

3. 1 Materials and Outcome Measures

The domain specific materials used in the study, which consisted of training materials for the simulation environment, pre/post test, and a 31 page booklet including introductory reading material about Rankine cycles and focused readings with suggested illustrative analyses to perform using the CyclePad simulator for three forms of Rankine cycles, were all developed by a Carnegie Mellon University mechanical engineering professor with the help of three of his graduate students and input from our team. These domain specific materials were exactly the same across conditions. Furthermore, all of the content presented in the set of dialogues the environment is capable of conducting with students in the Dynamic Support conditions is included in the reading materials in the form of “targeted text minilessons” [14], which present the same information using virtually identical phrasing, except that they are not interactive.

We computed two outcome measures using a Pre/Post test. 42 multiple choice and short answer questions were used to test analytical knowledge of Rankine cycles, including relationships between cycle parameters. An important aspect of this was a set of prediction questions where students were told to predict the impact of a specific change in one cycle parameter on several other cycle parameters. The other part of the test was a set of 8 open response questions assessing conceptual understanding of Rankine cycles. The third type of outcome measure we looked at was ability to apply knowledge to build and optimize a Rankine cycle using CyclePad during phases (4)-(6) of the experimental procedure. This was assessed by examining the simulator log files the students saved from the Implementation step (i.e., step 6) of the experimental procedure. Finally, we administered a questionnaire, developed in our prior work, only to students working in the Pairs conditions, which consisted of 8 questions meant mainly to evaluate how students felt about their collaboration experience.

3. 2 Experimental Procedure Common to All Condition

The experimental procedure consisted of 8 steps. (1) At the beginning of the lab session, students were lead through formal training on the simulation software from an instructor using power point slides and the simulation environment. (15 minutes). (2) The students then worked through the 32 page booklet, which presented to them the formal domain content instruction and instructed them to do specific activities with the simulation environment (70 minutes). (3) Subsequent to the formal instruction, but immediately prior to the experimental manipulation, students took the pretest (15 minutes). Steps (4)-(6) together constituted an exploratory exercise developed to allow students to apply the knowledge they learned during step (2). Immediately prior to step (4) of the experimental procedure, students were given instructions for steps (4)-(6) that described the terms of a “contest” to use what they just learned to design, build, and evaluate two Rankine cycle designs with the best thermal efficiency they can achieve. A first prize of a $25 gift

certificate and second prize of $10 were promised for the students who would achieve the highest and second highest average thermal efficiency between the two designs. For step (4), students were instructed to review what they learned and insert their notes into the chat buffer. The experimental manipulation took place during phase (4), which lasted for 25 minutes. (5) Next was the Planning/Synthesis step where students were instructed to review the notes generated during Step (4), and to form 2 concrete plans (10 minutes). (6) Finally, during the Implementation/Evaluation step, students were instructed to build their designs in the simulation environment and evaluate their thermal efficiency (25 minutes). (7) In order to assess what students learned from the exploratory exercise, they then took a post-test (15 minutes). (8) In the Pairs conditions only the students took the questionnaire.

3.3 Experimental Design

Our experimental manipulation consisted of 6 conditions resulting from a 3X2 full factorial design crossing a 3 level support factor (no support, static support, or dynamic support) with a two level collaboration status factor (working with a partner or working alone). The experimental manipulation took place during step (4) only, where students typed notes, ideas, and thoughts into an MSN like chat buffer as described above. In all other phases students worked independently in a consistent fashion across conditions. The instructions given to students in all conditions were identical except for the manipulation specific instructions inserted at the end of the instructions for step (4), which are described below. All students were instructed to review specific relevant portions of the book they had all been given, and which they had worked through during step (2) of the experimental procedure. Thus, the experimental manipulation only affected the students' experience interacting with the chat buffer.

3.4 Participants

We conducted our study over a four day period of time as part of a sophomore Thermodynamics course at Carnegie Mellon University. The week before the study, students were introduced to the topic of Rankine cycles during the lecture part of their course. They then completed a take home assignment related to Rankine cycles. Finally, they participated in a 3 hour lab session in which the experiment took place. Students were randomly assigned to conditions, subject to scheduling constraints. Each of 6 lab sessions were assigned to one of the 6 experimental conditions. 87 students participated in the study, including 13 students in each of the conditions where students worked alone, and 16 in each pairs condition.

4. Results

The clearest indication of the complementary value of collaboration and tutorial dialogue support come from the multiple choice portion of the test. In our analysis, we used an ANCOVA model with our two independent variables, where post test score was the dependent variable, and the pretest score was the covariate. We found a significant main effect in favor of the Pair condition $F(1,80) = 4.64$, $p < .05$, effect size .4 standard deviations. Furthermore, we found a significant main effect of the support variable $F(2,80) = 5.7$, $p < .005$, effect size .7 standard deviations. A pairwise Tukey test post-hoc analysis revealed that only the contrast between No Support and Dynamic Support was significant, with Dynamic Support being preferred. There was also a marginal interaction effect between the two independent variables $F(2,80) = 2.7$, $p = .07$. An analysis of the relative effect sizes of

the experimental conditions in comparison with the Individual-No Support condition reveals that the Pairs-Dynamic Support condition achieved the highest effect size, namely 1.24 standard deviations. Pairs with Static Support achieved an effect size of .9 standard deviations over the control condition. Individuals with Dynamic Support achieved an effect size of 1.06.

The conceptual portion of the test revealed an advantage for tutorial dialogue based support but not collaboration. As before, we used an ANCOVA model with our two independent variables, where conceptual post test score was the dependent variable, and the conceptual pretest score was the covariate. However, here we found only a marginal main effect of the support variable with the full ANCOVA model. With a simpler ANCOVA model where we drop the Pair versus Individual factor and only evaluate the effect of the Support variable we find a significant main effect of the support variable $F(2,83) = 3.07$, $p = .05$. A pairwise Tukey test post-hoc analysis revealed that only the contrast between No Support and Dynamic Support was significant, with Dynamic Support being preferred. The effect size is .5 standard deviations. Based on an informal analysis of the chat logs, it is not surprising that we only see an effect of collaboration on the multiple choice portion of the test since the majority of the conversations focused on content covered in this portion of the test.

The practical assessment revealed no differences between conditions. We examined student success on the Execution/Evaluation stage. 91% of students were successful at building one fully defined cycle, and 64% of students were successfully able to define two.

The questionnaire data in some ways corroborated the learning gains analysis but also alerted us to some issues that we must address in our ongoing work. We examined the results of the questionnaire using a repeated measures logistic regression model with Question Type (i.e., Knowledge, Help, Engagement, and Benefit) as a repeated measure, with each level corresponding to the subset of the questionnaire related to these specific aspects of the collaboration. Since students assigned a value between -5 and +5 as their answer to each question, we assigned a 1 to a student for a question category type if their average score across questions related to that category was above 0, and 0 otherwise. Additionally, we included the Support condition variable as well as Student nested within Support condition in the model. The R^2 value of the model was .35. There was no main effect of Support condition, although there was a main effect of Student, Question Type, and the interaction between Question Type and Support Condition (LR-Chi2 = 17.1, $p < .01$). Consistent with what would be expected based on the test results, students in the Dynamic Support condition rated themselves as benefiting significantly more from the collaboration. However, contrary to expectation, students in the Dynamic Support condition rated themselves and their partner as significantly less engaged in the material during the collaborative portion of the experimental manipulation than students in the other two support conditions. While the results on the test show promise that our Dynamic support mechanism can lead to learning benefits, consistent with the questionnaire data we noticed many comments in the chat logs that indicated that students were frustrated with the dialogue agent.

5. Conclusions

We have investigated the role of reflection in simulation based learning by manipulating two independent factors that each separately lead to significant learning effects. The important finding is that because the effect size achieved by combining the two treatments is greater than the effect achieved by either of the two treatments, we conjecture that each of these factors must contribute something different to student learning rather than being replacements for one another. Consistent with this, an informal analysis of the chat logs

reveals that the nature of the reflection students engage in with each other is distinct from that in connection with the dialogue agents. Specifically, while the dialogue agents lead students crisply through a reasoning process that connects observations with principles and ultimately with design decisions, student discussions were more intersubjective in nature. Equally important, an analysis of the chat logs and questionnaire data demonstrate that there is still room for significant improvement of our design for adaptive collaborative learning support. In our current work we are continuing to investigate the reasons for student frustration with the dialogue agents in order to develop a more effective approach.

Acknowledgements

This project is supported by ONR Cognitive and Neural Sciences Division, Grant numbers N000140410107 and N000140510043.

References

[1] Kollar, I., Fischer, F. & Hesse, F. W. (2006). Computer-supported cooperation scripts - a conceptual analysis. Educational Psychology Review.

[2]O 'Donnell, A. M. Structuring Dyadic Interaction Through Scripted Cooperation , in O'Donnell & King (Eds.) Cognitive Perspectives on Peer Learning, Lawrence Erlbaum Associates: new Jersey, 1999.

[3]Weinberger, A., Ertl, B., Fischer, F., Mandl, H. Cooperation scripts for learning via web-based discussion boards and videoconferencing. In Proc. The first joint meeting of the EARLI SIGs "Instructional Design" and "Learning and Instruction with Computers" Knowledge Media Research Center (2004), 22-28.

[4] Gweon, G., Rosé, C. P., Zaiss, Z., & Carey, R. (2006). Providing Support for Adaptive Scripting in an On-Line Collaborative Learning Environment, Proceedings of CHI 06: ACM conference on human factors in computer systems. New York: ACM Press.

[5] Weinberger, A., & Fischer, F. (2006). A framework to analyze argumentative knowledge construction in computer-supported collaborative learning. *Computers & Education*, 46, 71-95.

[6] Donmez, P., Rose, C. P., Stegmann, K., Weinberger, A., and Fischer, F. (2005). Supporting CSCL with Automatic Corpus Analysis Technology, in Proceedings of Computer Supported Collaborative Learning.

[7] Forbus, K. D., Whalley, P. B., Evrett, J. O., Ureel, L., Brokowski, M., Baher, J., Kuehne, S. E. (1999). CyclePad: An Articulate Virtual Laboratory for Engineering Thermodynamics. Artificial Intelligence 114(1-2): 297-347.

[8] Jordan, P., Ringenberg, M. and Hall, B. (2006). Rapidly Developing Dialogue Systems that Support Learning Studies. In Proceedings of ITS06 Workshop on Teaching with Robots, Agents, and NLP.

[9] Kumar, R., Rosé, C. P., Aleven, V., Iglesias, A., Robinson, A. (2006). Evaluating the Effectiveness of Tutorial Dialogue Instruction in an Exploratory Learning Context, Proceedings of the Intelligent Tutoring Systems Conference.

[10] Aleven, V, & Rosé, C. P. (2005). Authoring plug-in tutor agents by demonstration: Rapid rapid tutor development, Proceedings of AI in Education '05.

[11] Koedinger, K. R., Aleven, V., Heffernan, N., McLaren, B., & Hockenberry, M. (2004). Opening the Door to Non-Programmers: Authoring Intelligent Tutor Behavior by Demonstration. In J. C. Lester, R. M. Vicario, & F. Paraguaçu (Eds.), Proceedings of Seventh International Conference on Intelligent Tutoring Systems, ITS 2004 (pp. 162-174). Berlin: Springer Verlag.

[12] Rosé, C. P., Aleven, V., Carey, R., Robinson, A., Wu, C. (2005). A First Evaluation of the Instructional Value of Negotiatble Problem Solving Goals on the Exploratory Learning Continuum, Proceedings of AI in Eduction '05

[13] Wang, H. C., Rosé, C. P., Cui, Y., Chang, C. Y., Huang, C. C., & Li, T. Y. (2007). Thinking Hard Together: The Long and Short of Collaborative Idea Generation in Scientific Inquiry, Proceedings of CSCL 2007

[14] VanLehn, K., Graesser, A., Jackson, G. T., Jordan, P., Olney, A., Rosé, C. P., (in press). Natural Language Tutoring: A comparison of human tutors, computer tutors, and text. Cognitive Science.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Creating Contexts for Productive Peer Collaboration: Some Design Considerations

Amanda HARRIS[1], Rosemary LUCKIN[2] and Nicola YUILL[1]
[1] *IDEAs Lab, Department of Informatics, University of Sussex, UK*
[2] *London Knowledge Lab, Institute of Education, University of London, UK*

Abstract. Building technology enhanced learning contexts that work involves increasing our understanding of how we can model learners affective and cognitive profiles, the contexts within which they learn and the interactions between learner and context. In this paper we provide evidence that can inform the design of both models of learners and of their context. We explore the impact of children's goal orientation upon their collaborative interactions whilst completing computer-based tasks. This work suggests that mastery goals engender a willingness to engage in the process of argumentation and discussion that is an important feature of effective collaborative learning. By contrast, performance goals lead learners to try and demonstrate their ability using their collaborative partner to affirm their own competence. This work informs the design of motivational learner models in collaborative contexts. In addition, we found that it is possible to encourage a particular goal orientation in a learner through manipulations to the task presentation, and in so doing to engender more productive styles of collaborative interaction. This finding informs the design of the collaborative context.

1. Introduction

Research from psychology and education shows that working collaboratively in pairs or small groups can have positive effects on children's learning and development. However classroom observations suggest that the quality of children's collaborative discussion tends to be poor; while children may be seated in groups and share resources, the quality of their discussion tends be low [1]. Much of their experience of working collaboratively occurs in the context of computer-mediated activity which typically involves pairs or small groups working around a single computer [2]. The frequency of this activity and children's difficulties with effective interactions has provided a natural context for development of systems designed for face to face classroom collaboration. However, in order for technology to support the *development* of collaborative skills we need to know more about when and why children's collaborative activity is more or less productive.

In this paper we identify an approach to modelling and supporting collaborative interaction by focusing on the motivational context of the activity using achievement goal theory as a framework. First we discuss children's collaborative interactions and the potential influence of goal orientation before presenting findings from two empirical studies. Finally, we outline some design considerations for systems which might use goal theory to support children's face-to-face collaboration.

2. Children's computer-mediated collaborative learning

Definitions of collaborative activity centre on the notion of mutuality and joint activity [3] and have in common a view of collaboration as reciprocal interaction in which ideas and perspectives are exchanged and explored and solutions are reasoned and justified. For example, Roschelle & Teasley [4] define collaboration as 'a coordinated, synchronous activity that is the result of a continued attempt to construct and maintain a shared conception of a problem' (p. 70). Productive interaction is therefore associated with the ability to engage in effective discussion by providing reasons, explanations and justifications for ideas and suggestions.

Without significant support or training the type of discussion outlined above is often lacking in children's spontaneous classroom interactions. Factors such as group size, task design, gender and ability all impact on the likelihood of peer collaboration being productive and having a positive learning outcome [2]. For example, primary-age children tend to work more effectively with same-gender age mates and in smaller groups such as pairs or threesomes. Also the way in which collaboration is represented, through for example interface design [5] and hardware configuration [6] has an influences the quality of discussion and nature of engagement. However, even when 'ideal' conditions are engineered, interactions between children differ markedly; some children consistently fail to work together productively [7].

We argue that research has tended to focus on the *objective* differences between children. By objective, we mean variables over which the child has little control. For example, gender, ability, the type of task and the size of the group are, for the duration of the collaboration, not in the child's direct control. The *subjective* agency learners bring to the collaborative group, such as their willingness and attitude towards group work, has for the most part been neglected [8]. However, in outlining some design considerations for collaborative technology for classrooms Scott et al. (2003) state that:

'Multiple interaction styles should be supported in both hardware and software: This will allow children to explore a variety of collaborative strategies and choose the most suitable one(s) for the activity and their personalities' (p. 226).

This moves some way toward considering children's subjective agency and places a focus on the characteristics the children themselves bring to a collaborative exchange. Before technology can be designed with this is mind, research needs to identify the characteristics which are associated with different interaction styles. This involves identifying behavioural differences as well as the factors which underlie and influence such behaviours. In considering these questions we focus here on children's achievement goal orientation as a means of understanding different learning behaviours as well as underlying cognitive and affective responses to learning.

3. Achievement motivation and collaborative learning

Achievement motivation theorists argue that the achievement goals an individual holds determine their beliefs about and attitudes towards learning and consequently their learning behaviour. The emphasis for achievement goal theorists is therefore not on the quantity of an individual's motivation but rather on the quality of that motivation. Goal orientations are typically characterised according to two patterns of goal adoption. Mastery goals orient learners towards developing competencies by mastering new skills and understanding new concepts [9]. They are associated with high levels of task

engagement, the use of self-regulated learning strategies, effort and persistence [10]. Performance goals, on the other hand, are underpinned by the desire to demonstrate ability in comparison to others. These goals are associated with more surface-level learning strategies, the avoidance of challenging tasks and a concern with comparative evaluations of ability [9, 10]. Given these different approaches to learning there is a strong theoretical basis for predicting that achievement goals will influence collaborative behaviour in distinct ways and may therefore account for some of the variation in the quality of interactions observed in the classroom.

Using Rochelle and Teasley's [4] definition of collaboration a child motivated by performance goals may find 'coordinated activity' difficult as this inevitably minimises individual markers of achievement in favour of group progress. The social comparative nature of a performance orientation may, therefore, lead the child to perceive a collaborative context as either a forum for the public display of individual ability or a threatening environment in which a lack of ability may be exposed. This represents potential conflicts for the child between the demands of a collaborative task and individual perceptions of what might constitute success on that task.

On the other hand, the mastery-oriented child may find a collaborative context as more appealing in that they are more likely to view peers as sources of information which aid task mastery rather than as a threat to their perceptions of ability [11]. The high levels of task engagement associated with mastery goals, regardless of perceived ability [10] may enable mastery-motivated children to engage with and participate in collaborative learning activities more productively. Mastery goals are also associated with the use of metacognitive and self-regulatory learning strategies [9, 10]. The mastery-oriented child may, therefore, be at an advantage as collaborative exchanges challenge children's metacognitive abilities.

4. Empirical Studies

While the link between collaborative styles of interaction and achievement goal orientation seems intuitively and theoretically valid, there is little research which connects these literatures empirically. Although there is recent work emerging in relation to undergraduate students [12], in general and particularly in relation to children, the collaborative learning literature has tended not to address motivational influences on behaviour. In the same vein, achievement goal research has not been specific about which behaviours are relevant to particular learning contexts.

In the light of this we undertook two studies which, although they used different methods, had a common aim; to identify the influence of achievement goals on children's behaviour when collaborating with a peer. Both studies involved free behavioural observation of children's interactions. This distinguishes our approach from other related achievement goal work which tends to rely on learners' self-reports after a particular learning experience [For example, 12]. In both studies we observed same-gender pairs of primary-aged children (7 to 10 years old) interacting around a desktop computer using a single input device; a configuration children are most familiar with in the classroom. Each child's contribution to discussion was considered in relation to individual differences in goal orientation (Study 1) and contextual differences in an emphasis on mastery or performance goals (Study 2). In drawing together the results from both studies we will present a mastery and a performance profile of collaborative behaviour.

4.1. Study 1

Twenty-two children from two urban primary schools in the South East England participated in this study. The children were divided into 11 same-gender, same-school pairs and observed using a software system called Joke City. The software prompts reflection on language ambiguity and humour in jokes and riddles. Two video recorded interactive sessions per pair, each between 15 and 30 minutes long, were analysed. Our coding scheme distinguished between those behaviours we identified as indicating productive interaction (e.g., *descriptions, justifications, disagreements)* and those indicating less productive interaction (e.g., *commands, submissions, dismissing and off-task behaviour*) as well as comments indicating metacognitive awareness and regulation (e.g., *metacognitive description of self, other and joint understanding*). The proportional frequency of language categories for each child was summed and correlated with mastery and performance scores measured using a teacher-rated version of the Patterns of Adaptive Learning Scales (PALS) questionnaire [13].

4.1.1. Results

The achievement goal scale yielded a mastery and a performance score for each child. The overall mean mastery score was 3.14 (SD .72) and the overall performance mean was 2.9 (SD .46). Our analysis revealed that mastery goals were positively correlated with disagreements ($r = .58$, $p < 0.01$) and negatively correlated with submissions ($r = -.38$, $p < 0.05$). Submissions involved children giving in to a partner without indicating understanding. We also found a relationship between performance goals and metacognition; performance goals were positively related to references to the child's awareness of their own knowledge or understanding ($r = .51$, $p <0.05$) while negatively related to references to their partner's knowledge or understanding ($r = -.49$, $p <0.05$).

These results are consistent with some of the key characteristics associated with mastery and performance goals identified in the literature and suggest which behaviours in particular, within a general mastery or performance profile, are relevant within a collaborative context. However, the range of mastery and performance scores measured using the PALS questionnaire suggested that neither goal orientation was particularly strong. We now turned to examining the extent to which it might be possible to manipulate goal orientation and foster a particular style of interaction.

4.2. Study 2

In this study we examined the influence of goal-oriented contexts on children's collaborative behaviour while playing a computer-mediated logical reasoning game called Zoombinis [14]. The game consists of a series of puzzles which players have to solve by using thinking skills such as identifying patterns and reasoning about evidence. This is done by combining and organising the Zoombini's features (e.g., hair colour, type of footwear) and different features of the environment.

In order to assess the influence of goal-oriented contexts we distinguished between children who had strong goal orientations from those who appeared neutral and therefore may be open to external motivational cues [15]. From a sample of 61 children between 8 and 10 years, we identified 34 who were neutral in terms of goal orientation and were suitable for pairing. These children were matched in same-

gender, same-class pairs (all children in this study came from a single primary school in a city in the South East of England). Eight pairs were randomly assigned to a mastery-focused condition and 9 to a performance-focused condition.

Each pair was first shown how to use the software (e.g., help and map icons were explained). All pairs were then given instructions to work together and discuss the puzzles as much as possible. After these initial instructions pairs then received different instructions about the object of the game depending on which condition they had been assigned. In the mastery-focused condition pairs were told that the object of the game was to work out good strategies for solving the puzzles and that it did not matter if they lost any Zoombinis in trying to achieve this aim. In the performance-focused condition pairs were told that the object of the game was to get as many points as possible and that each Zoombini was worth one point. Pairs were observed playing Zoombinis during a single session for approximately 25 minutes in length.

With reference to both the achievement goal and collaborative learning literatures we extended our coding scheme to focus specifically on behaviours which might characterise a collaborative interaction as being mastery or performance oriented. We were interested in three types of participation. Firstly, we distinguished between complex problem solving which included reasons, justifications and explanations and simple problem solving which was merely procedural or descriptive, providing suggestions without elaboration. Secondly, we measured expressions of awareness of knowing and thinking and distinguished between metacognitive awareness of held knowledge or understanding (*e.g., I know*) and awareness of a lack of knowledge or understanding (*e.g., I don't know*). Thirdly, we were interested in exploring differences in the specific metacognitive strategy of help-seeking.

4.2.1. Results

Analysis was conducted by calculating proportional frequencies of each language category for each participant. In order to explore whether there were any differences in problem-solving language we conducted a mixed ANOVA with problem-solving type (simple and complex) within subjects and goal orientation between subjects. This revealed a main effect of problem solving type with children making more simple problem solving utterances overall ($F(1.\ 30) = 432.41$, $p < 0.001$). However there was a significant interaction between goal orientation and problem solving ($F\ (1, 30) = 5.36$, $p < 0.05$) where children in the mastery-focused condition made significantly more complex problem-solving utterances than children in the performance group. In relation to metacognitive awareness children in the performance group made more metacognitive positive comments (e.g., *I know*) (Mean = 9.6, SD = 4.5) than children in the mastery group (Mean = 7.4, SD 5.02). However, the large standard deviations suggest this was highly variable and the difference between groups did not reach significance. In relation to help seeking children sought help either by accessing the help function provided by the game (task-focused) or by requesting help from the researcher (external help). A count of each type of help for each child was measured. Chi square analysis revealed that children in the performance group were significantly more likely to use external help than children in the mastery group ($x^2(1) = 7.56$, $p < 0.01$). There was no difference between groups in children's use of task-focused help.

5. Discussion

In both studies, mastery goals were associated with behaviours more conducive to productive interaction and more likely to promote learning. In Study I, the stronger children's mastery goals were, the more they engaged in constructive disagreements and the less they tended simply to submit to their partner's suggestions. In Study II, mastery goals were associated with complex problem-solving which involved providing justifications and explanations for suggestions. In both studies therefore mastery goals appeared to engender a willingness to engage in a process of argumentation and discussion. The core feature of a mastery orientation is identified as the desire to gain understanding and master tasks without concern for making public mistakes or exposing a lack of ability [9, 10]. When children are collaborating, this motivation becomes evident in the level of the complexity of their discussion. Providing explanations and justifying one's perspective will necessarily carry with it the risk of exposing an incorrect solution or understanding of the problem. However, it is through making one's understanding explicit and discussing alternative perspectives that new insight is gained in the course of interaction [16]. Holding mastery goals seems to complement and support this process.

Similarities between studies suggest that a performance orientation may be associated with an individualistic style of peer interaction. In Study I, we found that performance goals were related to types of metacognitive talk which indicated a focus on the self. The stronger a child's performance goals the more likely they were to make self-referring metacognitive utterances and the less likely they were to refer to their partner's knowledge or understanding. In Study II, children in the performance goal group also used a higher number of self-referring metacognitive statements. These results suggest that holding performance goals may inhibit a child's ability to engage in the coordinated, joint activity associated with productive collaborative interaction [4]. A core element of a performance orientation is the desire to demonstrate ability[9, 10]. Collaborative partners may therefore provide the performance-oriented child with the opportunity to affirm their own competence.

6. Implications for design

The above discussion suggests two issues for designing technology for collaboration in the classroom. Firstly, children will behave differently during collaborative interactions depending on whether they are pursing mastery or performance goals. Secondly, most children are very responsive to goal-oriented cues embedded in the learning context. These findings suggest that a) it is crucial to consider goal orientation when constructing a model of the learner and b) using goal-oriented cues in the design of the collaborative context may encourage more adaptive motivational approaches.

6.1. Modelling collaborative learners

A collaborative learning context poses a challenge to any teacher, either human or machine, as the characteristics of not one but two or more learners need to be considered simultaneously. The machine teacher is at a further disadvantage given the constraints of technology to 'listen' to the verbal exchange that is fundamental to such

a learning context. With this limitation in mind we suggest three categories of goal-oriented behaviours which a system could detect and monitor:

- *Help-seeking*: Children pursuing performance goals will tend to look for help that offers solutions and also provides reassurance.
- *Complex problem-solving*: Children pursuing mastery goals tend to engage in more complex problem-solving in which they provide justifications and explanations for their suggestions. One approach which could be applied and which has been borrowed from asynchronous collaborative environments, is to provide sentence openers for participants to select in order to give the system an indication of the nature of the discussion [17].
- *Disagreements*: Mastery goals are more likely to be related to constructive disagreements. Recent developments in how collaboration is represented through technology, for example through interface design and hardware configuration [5], affords the possibility of monitoring both individual activity (through seperate input devices) as well as the process through which individual activities are coordinated (interface representations of agreements and disagreements).

6.2. Designing collaborative contexts

We identified earlier the difficulties many children have with engaging constructively in collaborative activity. In order to build technology that works, we therefore need to design not merely for collaborative use but also in a manner that guides the development of collaborative skills. The results of our classroom observations suggest that some difficulties may be associated with the pursuit of performance goals for achievement while mastery goals promote productive interaction. Our experimental manipulations also suggest that it is possible to encourage children to adopt particular goals through constructing goal-oriented contexts. We suggest that fostering a mastery context could be achieved through techniques such as:

- *Motivational instructions for the task*: Inherent in our instructions were messages about the purposes and meaning of the collaborative task. If we want to create systems which encourage children to engage collaboratively in deep learning, it would seem more appropriate to foster environments in which learning itself is the goal rather than receiving external rewards.
- *Scaffolding goal-oriented contexts:* The emphasis of scaffolding prompts may be designed to highlight particular goals for the interaction. For example, feedback and assistance which emphasise understanding, problem-solving and learning may be preferable to feedback which focuses on praise for ability or which is solution focused. There is also the potential to use goal orientation as an additional level of fading within a more general scaffolding structure. A method which has the possibility of integrating personal goals (detected through a learner model) and contextual goals (provided by the software scaffolding).

7. Conclusions

Interactive learning environments are increasingly being designed with an emphasis on the motivational and affective state of the learner [18]. In this paper we have presented

evidence which suggests that the achievement goals children pursue are crucial to consider within this broader framework. Mastery and performance goals influence collaborative behaviour in distinct ways; mastery goals promote collaborative discussion and argumentation while performance goals encourage an individualistic style of interaction. We also highlight the role of the learning context in shaping the sorts of goals children pursue. This suggests that motivational aspects of context can be designed in such a way to support the development of children's collaborative skills.

8. References

[1] Baines, E., P. Blatchford, and P. Kutnick, (*2003) Changes in grouping practices over primary and secondary school.* International Journal of Educational Research, 2003. **39**: p. 9-34.
[2] Light, P. and K. Littleton, (1999) *Getting IT together*, in *Learning with Computers*, K. Littleton and P. Light, Editors. 1999, Routledge: London.
[3] Goos, M., P. Galbraith, and P. Renshaw, (*2002) Socially Mediated Metacognition: Creating Collaborative Zones of Proximal Development in Small Group Problem Solving.* Educational Studies in Mathematics, 2002. **49**: p. 193-223.
[4] Roschelle, J. and S. Teasley, (1995) *The Construction of Shared Knowledge in Collaborative Problem Solving*, in *Computer Supported Collaborative Learning*, C. O'Malley, Editor. 1995, Springer: Berlin.
[5] Kerawalla, C., et al., (*in press) "I'm keeping those there, are you?" The role of a new user interface paradigm - Separate Control of Shared Space (SCOSS) - in the collaborative decision-making process.* Computers and Education, in press.
[6] Scott, S.D., R.L. Mandryk, and K.M. Inkpen, (*2003) Understanding children's collaborative interactions in shared environments.* Journal of Computer Assisted Learning, 2003. **19**: p. 220-228.
[7] Barron, B. and D. Sears, (2002) *Advancing understanding of learning in interaction: How ways of participating can influence joint performance and individual learning*, in *Computer support for collaborative-learning: Foundations for a CSCL community*, G. Stahl, Editor. 2002, Erlbaum: Hillsdale, New York.
[8] Schwartz, D.L., (1999) *The Productive Agency that Drives Collaborative Learning*, in *Collabortive Learning: Cognitive and Computational Approaches*, P. Dillenbourg, Editor. 1999, Elsevier Scienve Ltd.: Oxford.
[9] Ames, C., (*1992) Classrooms: Goals, Structures, and Student Motivation.* Journal of Educational Psychology, 1992. **84**(3): p. 261-271.
[10] Dweck, C.S. and E.L. Leggett, (*1988) A Social - Cognitive Approach to Motivation and Personality.* Psychological Review, 1988. **95**(2): p. 256-273.
[11] Blumenfeld, P.C., (*1992) Classroom learning and motivation:clarifying and expanding goal theory.* Journal of Educational Psychology, 1992. **84**(3): p. 272-281.
[12] Darnon, C., et al., (*2006) Mastery and performance goals predict epistemic and relational conflict regulation.* Journal of Educational Psychology, 2006. **98**(4): p. 766-776.
[13] Midgley, C., et al., (2000) *Manual for the Patterns of Adaptive Learning Scales (PALS)*. 2000: Ann Arbor, MI:University of Michigan.
[14] Hancock, C. and S. Osterweil. (1996) *Zoombinis and the Art of Mathematical Play*. 1996 [cited 2006 25 May]; Available from: http://www2.terc.edu/handson/s96/zoom.html.
[15] Harris, A., N. Yuill, and R. Luckin, (*under review) Examining the consistency of mastery and performance goals across group and perceived-ability contexts.* Learning and Individual Differences, under review.
[16] Wegerif, R., N. Mercer, and L. Dawes, (*1999) From social interaction to individual reasoning: an empirical investigation of a possible socio-cultural model of cognitive development.* Learning and Instruction, 1999. **9**: p. 493-516.
[17] Robertson, R., J. Good, and H. Pain, (*1998) BetterBlether: The Design and Evaluation of a Discussion Tool for Education.* International Journal of Artificial Intelligence in Education, 1998. **9**: p. 219-236.
[18] du Boulay, B. and R. Luckin, (*2001) Modelling human teaching tactics and strategies for tutoring systems.* International Journal of Artificial Intelligence in Education, 2001. **12**(3): p. 235-256.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Who Says Three's a Crowd? Using a Cognitive Tutor to Support Peer Tutoring

Erin WALKER[1], Bruce M. MCLAREN[1], Nikol RUMMEL[2], Ken KOEDINGER[1]
[1]*Human-Computer Interaction Institute, Carnegie Mellon University, Pittsburgh, USA*
[2]*Institute of Psychology, University of Freiburg, Freiburg, Germany*
{erinwalk, koedinger, bmclaren}@cs.cmu.edu; rummel@psychologie.uni-freiburg.de}

Abstract. Adding student collaboration to an intelligent tutoring system could leverage the benefits of both approaches. We have incorporated a mutual peer tutoring script, where students of similar abilities take turns tutoring each other, into the Cognitive Tutor Algebra. In this paper, we identify three design principles for peer tutoring, and discuss how they were realized in our peer tutoring script. We then develop a cognitive model for peer tutoring, and drawing from student data, identify places for an intelligent tutor to provide feedback. Finally, we describe the implementation of the script and our plans for formal evaluation.

Keywords. Peer tutoring, intelligent tutoring systems, collaborative learning

1. Introduction

Augmenting an intelligent tutoring system (ITS) with collaborative learning activities holds the promise of increasing the benefits of the ITS. The structured problem-solving and individualized feedback provided by an ITS such as the Cognitive Tutor Algebra (CTA) has increased student learning by approximately one standard deviation over traditional classroom instruction [1]. Critics of this approach argue that students acquire shallow knowledge while using the ITS, since there is not much freedom for students to construct their own knowledge or learning path. On the other hand, collaboration has also been shown to have positive effects on learning, and group activities encourage students to develop deep knowledge [2]. However, these activities are only effective at increasing learning if students interact in positive ways [3]. Students typically do not collaborate well spontaneously and often do not receive enough guidance from their teacher. The CTA curriculum includes collaborative work, but these activities are often not optimally administered by teachers or followed by students [4]. To facilitate appropriate interaction, researchers develop collaboration scripts in which participating students are given specific roles and activities [5]. An ITS can provide a platform for implementing a script and adaptive support for students following the script.

If ITS methods can be applied to tutoring collaborative activities, the benefits of both approaches could be combined and strengthened. Some preliminary work has been done in this area: Harrer et al. [6] have integrated the collaborative tool Cool Modes with the Cognitive Tutor Authoring Tools (CTAT), an authoring tool for tutoring systems, and Soller [7] has created a model of structured chat in order to

develop an ITS to improve student interaction. Another approach is to use an intelligent agent as one of the collaborators. Biswas et al. [8] had students, helped by a mentoring agent, teach an intelligent agent about ecosystems. Our approach involves incorporating peer tutoring into the CTA. We designed a peer tutoring addition that draws from previous successful peer tutoring scripts, and based on a task analysis and empirical data, developed a preliminary model for peer tutoring and adaptive cognitive support. We have implemented the extension and will soon be conducting a formal evaluation.

2. Peer Tutoring Addition to the Cognitive Tutor Algebra

In our peer tutoring addition to the CTA, we adopted a reciprocal peer tutoring script, where students of similar abilities take turns tutoring each other on course material. This type of peer tutoring has been shown to increase mathematics learning in a realistic classroom environment [9]. In order to explore the factors that make peer tutoring an effective learning intervention for both the tutor and the tutee, researchers have scripted the peer tutoring process to encourage students to behave in particular ways, and then compared the scripted condition to an unscripted control. In conditions where tutors are encouraged to provide elaborated explanations [10], set goals for tutoring and monitor the skills being acquired [11], and prepare ahead of time [9], peer tutors tend to learn more. Researchers have also explored the effect of the peer tutor on the tutee by coding tutoring transcripts for particular behaviors, and correlating those with learning. Tutees learn more when they request help, tutors respond with deep explanations, and tutees apply the explanations to their problem-solving [12].

Biswas et al. [8] identified three aspects of interaction that exist in learning by teaching: taking responsibility for actions, reflecting on knowledge, and developing structured knowledge. These processes are also present in successful peer tutoring. When students prepare, they take responsibility for knowledge because they will soon be communicating it to another. Students also must monitor their tutee's knowledge, becoming more aware of problem skills and identifying gaps in their own knowledge. Lastly, during tutoring, students must ask questions and receive explanations, leading them to better structure their knowledge. From this empirical and theoretical literature, we have derived three design principles for our peer tutoring script. Before tutoring, students should complete exercises on both the skills required to solve domain problems and the skills required to teach them (*preparation*). As a tutor, students should set goals for their partner and monitor their partner's progress (*reflection*). Students should talk about the domain in elaborated and specific ways (*interaction*).

Our peer tutoring script is built on an equation solving unit of the CTA, where students are given a prompt (e.g., "Solve for x") and an equation (e.g., "ax + bx =c"). To implement the *Preparation* principle, we divided student activity into two phases: a preparation phase, where students solve problems using the CTA, and a collaboration phase, where students tutor each other on the problems they solved. In the preparation phase, students solve problems using an equation solver tool and receive feedback from the cognitive tutor. Their skill mastery is displayed in a "skillometer". After each problem, they read descriptions of good tutoring behavior and then answer questions such as "What would be a good hint to give to your partner?"

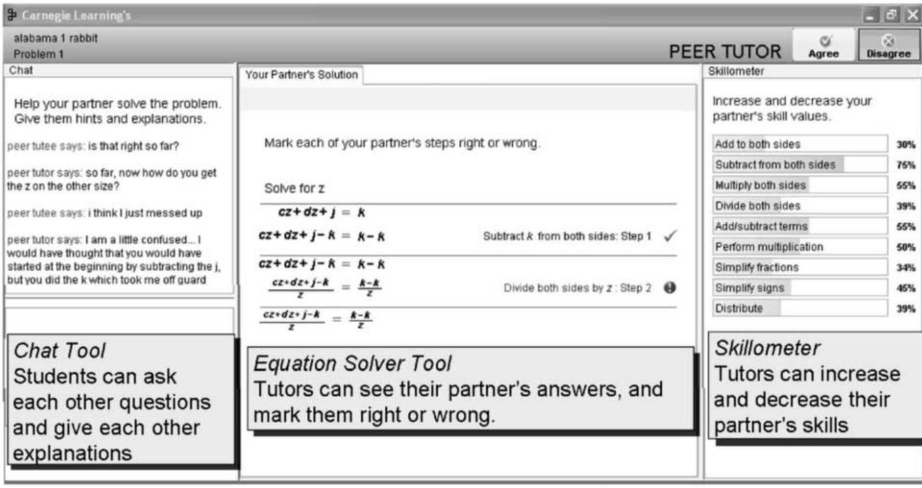


Figure 1. Peer Tutor Interface

During the collaboration phase, students are put in same-ability groups and collaborate at different computers in the same classroom. They take turns being the tutor and tutee. See Figure 1 for a screenshot of the peer tutor's interface. The tutee still solves the problem with the equation solver, as if using the regular ITS. Peer tutors can see the tutee's actions, but cannot solve the problem themselves. Instead, to incorporate the *Reflection* principle, the peer tutor can mark the tutee's answers in the equation solver and adjust the tutee's skills in the skillometer. These actions, seen by the tutee, prompt students to reflect on skills required to solve the problem and how close the tutee is to mastering those skills. Students also discuss the problem in a chat window, following the *Interaction* principle.

As an illustration of our approach, the following is an excerpt from a real tutor-tutee interaction. The students were solving the problem "$cz + dz + j = k$" for z. The tutee subtracts k from both sides in the equation solver, and then asks in the chat, "Is that right so far?" The tutor can see the tutee's action, and responds incorrectly: "So far, now how do you get the z on the other side?" The tutee divides by z, and realizes something is wrong, typing "I think I just messed up." The tutor responds, "I am a little confused… I would have thought that you would have started at the beginning by subtracting the j, but u did the k which took me off guard." The tutor then marks the step wrong in the equation solver. The tutee, seeing the tutor's feedback, undoes the incorrect steps and takes the correct step, subtracting both sides by j. The tutor is satisfied, and increases the value of the "Subtract both sides" skill bar. Once the students have solved the problem, the tutee clicks the done button, the tutor agrees that they are done, and the students move to the next problem and switch roles.

This interaction comes from a pilot study in which we explored how students used the script without any adaptive support (see [13]). We can use this data to determine where adaptive support might be most effective during peer tutoring. The script was implemented as an addition to the CTA, and piloted using 20 students from two Algebra-1 classes. We found that students did learn from the peer tutoring, and appeared to be engaged in preparing, interacting, and reflecting on their knowledge. However, peer tutors struggled to provide tutees with answers, despite having already solved the problems and having been given printouts of the answers. They did not complete many problems, skipped problems without completing them, and relied too heavily on teacher assistance. These behaviors are undesirable because no matter how

well students are collaborating, fewer problems successfully completed means fewer opportunities to master domain skills. Targeting intelligent tutoring support towards the cognitive aspects of peer tutoring may provide students with greater benefit from positive interaction, as well as leading to more problems successfully completed.

3. Cognitive Model for Peer Tutoring

Because of the results of the pilot study, we decided to focus our attention on providing adaptive support for peer tutors in knowing how to solve the problem they are tutoring and being able to communicate that knowledge to their partner. Further supporting the design principles that we identified, as do many other models of peer tutoring incorporated in peer tutoring scripts, would not be helpful without first ensuring that the peer tutor is capable of correctly advising the tutee.

3.1.Rational Task Analysis

To evaluate whether peer tutors are doing their job well and to support them in performing the tutoring task, we have designed a model of peer tutoring (Figure 2). For simplicity, our model is constrained in three ways. First, it assumes that peer tutor and tutee actions are synchronous, in that every action by the tutee is followed immediately by a tutor response. In practice, this is unlikely and may be undesirable; a fast tutee should not be held up by a slow tutor. Second, our model assumes that certain actions will always be taken even if those actions have been made redundant by other actions. For example, we have the peer tutor mark an answer incorrect before giving the peer tutee feedback. In a real situation, the peer tutor might simply tell the tutee that their answer is incorrect while giving them feedback, and therefore would not need to explicitly mark it. Our model treats this single action as two separate actions. Finally, our model focuses on the steps of the tutorial process, rather than the content. We are not modeling what should occur during discussion, but *when* discussion should occur. Once we can successfully support the tutoring process, we will focus on the content.

The model starts when the students are in a state of "working on the problem". The peer tutee can take one of three actions: take a problem step, ask for help from the peer tutor, or indicate they are done. The peer tutor can also start the model by determining that the peer tutee needs help. The model can then be divided into three general types of peer tutor responses: correction (bold boxes), skill assessment (dashed boxes), and discussion (regular boxes). The model assumes that correction is the immediate response to the tutee taking a step or selecting done. If the tutee action is correct, the tutor should mark it right. If the step or action is incorrect, the tutor should mark it wrong. The peer tutor starts a discussion whenever the tutee needs help or feedback, which may be the case after an incorrect answer by the tutee, when the tutee requests help, or when the tutee appears to be struggling. We have divided the discussion into three substeps: Initiation ("Tutor starts discussion"), the bulk of the discussion ("Students discuss step"), and termination ("Tutee understands?"). The peer tutor engages in skill assessment after correction and discussion actions, adjusting the tutee's skill bars as appropriate. Skill assessment is the lowest priority; the tutee needs feedback in order to continue but can move to the next step as the tutor adjusts skills.

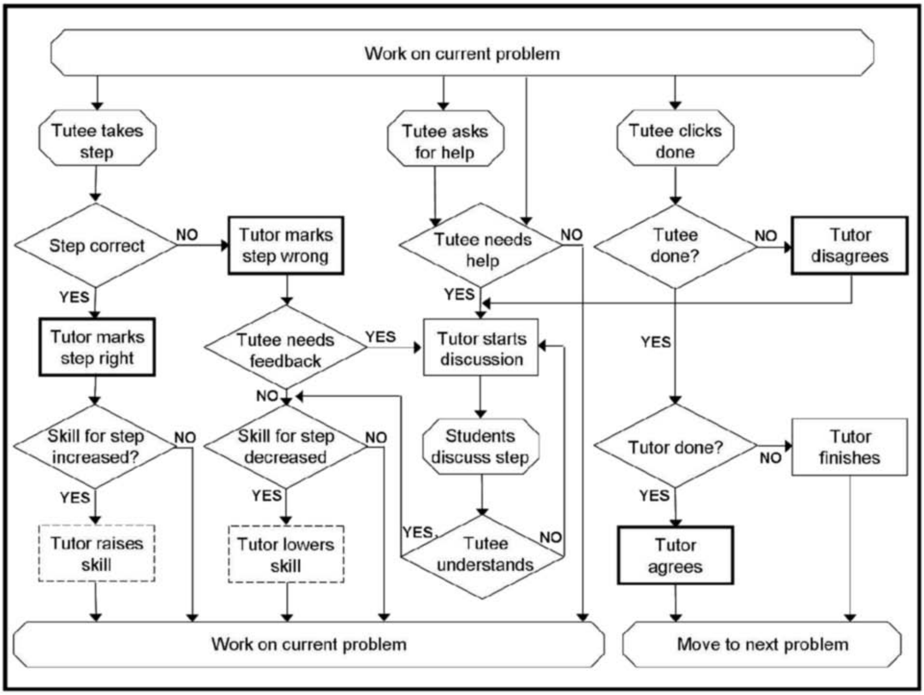


Figure 2. "Ideal" model of peer tutoring. Bold boxes are correction actions, dashed boxes are skill assessment actions, and regular boxes are discussion actions.

3.2. Design of Hints and Feedback

Drawing from the data collected in our pilot [13], we designed preliminary ITS support for peer tutoring based on two principles. We noticed a lot of interaction in the chat window between the peer tutor and the tutee; when stuck, tutees would use the chat to ask tutors for help, and tutors would use the chat not only for explanations, but to provide tutees with feedback on whether steps were right or wrong. To avoid disrupting this interaction, communication between the peer tutee and the computer tutor is mediated by the peer tutor. To get a hint, the tutee requests one from the peer tutor, and the peer tutor then requests one from the computer tutor. Any feedback messages given by the computer tutor are displayed to the peer tutor, who then explains the message to the tutee. For similar reasons, the computer tutor provides feedback to the peer tutor based on action, not inaction. If the peer tutor marks a wrong answer right, the computer tutor will step in, but if the peer tutor fails to correct an action, the computer tutor will not intervene, as it is possible that students are having a productive discussion or the peer tutor is occupied with another problem step. These hints and feedback are designed to help students complete problems correctly, so that they will benefit more from positive interaction (following the *Interaction* principle).

In our pilot, students in one class completed on average less than five problems during the collaborative session, while students in the other class only completed roughly 60% of the problems they attempted. Providing feedback on peer tutor correction actions might help students to correctly complete more problems. When peer tutors are wrong in a correction action, it is highlighted in the interface, giving them a visual indication that they have made a mistake. If peer tutors mark a correct problem

step incorrect, they get a message like, "Your partner has the right idea. Ask them why they took that step." If peer tutors mark an incorrect problem step correct, he or she gets a more complicated message involving the computer tutor feedback the tutee would have received (e.g., "It is better to divide by -7.1, since that would leave *y* and not –*y*"), and a prompt to collaborate like "Explain to your partner what to do and why." Similarly, if peer tutors agree with an incorrect tutee done action, they are informed that the action is incorrect, and students are not moved to the next problem.

During the pilot, we noticed that peer tutees would ask tutors for hints, and tutors would be unable to provide guidance, saying "I don't know" in response to questions or waiting for teacher help. To give the peer tutor extra assistance, we now allow them to ask the computer tutor for a hint. The hint, which has a cognitive and collaborative component, is then delivered only to the peer tutor. The cognitive component involves the advice the computer tutor would have given to the tutee during a typical tutoring session at the problem step the tutee is currently on, such as "-7.1*y* is -7.1 times *y*. How do you undo multiplication?" If the computer tutor advice is multi-level, the current hint is multi-level. The collaborative component involves a prompt to discuss feedback with their partner, such as "Talk to your partner about how this hint applies to the problem."

Finally, during the pilot, we noticed that peer tutors would tend to raise their partner's skill bars all the way to the maximum during the first or second problem. In an attempt to encourage better reflection on problem skills (following the *Reflection* principle), we send the tutor a feedback message if they try to raise a skill bar more than 15% per problem that reads, "Slow down! Before increasing more, wait until your partner has shown this skill on another problem." Students receive a similar message if they try to lower a skill bar more than 15%. In the future, we hope to have skill bar feedback that tutors more specifically which skill the peer tutor manipulates. We also intend to tutor the chat discussion. However, we believe the correction tutoring we have designed will be effective in itself at supporting peer tutors as they instruct their tutees.

4. Implementation of Peer Tutoring Script with Cognitive Tutoring Support

To implement the peer tutoring script, we modified the standard CTA architecture. The CTA separates the interface to the user (*tool*) from the intelligence (*tutor*) and from a central student model (*learner management*). We changed this architecture to enable multiple tool and tutor components. With this multi-tool capacity, the peer tutor and tutee can interact using separate interfaces at separate computers. With the capacity for multiple tutors, other tutoring modules such as a collaborative tutor can be added (see Figure 3a). To enable multiple components in different configurations, we changed the standard CTA components to function independently and remotely (see [14]). We then expanded the control module of the CTA to include a *mediator,* which intercepts all messages sent by a component and directs them to appropriate targets. The mediator maintains a list of session types, containing tools, tutors, and instructions for how messages are passed. The most basic session type contains the standard CTA tool and the cognitive tutor. The peer tutoring session type has a peer tutee tool, a peer tutor tool, and an echoing agent which facilitates collaboration by echoing a user action on one tool to the other tool's interface.

The first step to incorporating computer tutor feedback into our addition to the CTA was to develop a separate tutor module, called the correction tutor. The correction

tutor has knowledge of peer tutoring actions instead of domain knowledge. The expert model of the correction tutor is based on a simple bug rule: If the peer tutor response to a tutee step does not match the cognitive tutor response, the bug rule fires, and the correction tutor highlights the problem step and displays feedback on the peer tutor's screen. The peer tutor must have responded to the step before the rule can fire, and thus the model does not force the peer tutor to respond to every step. The model also has a bug rule to deal with overzealous manipulation of skill bars.

Figure 3a. High-level overview of multi-component configuration. Dashed components are only found in the script with ITS support.

Figure 3b. Message passing logic in the mediator. Dashed components are only found in the script with ITS support.

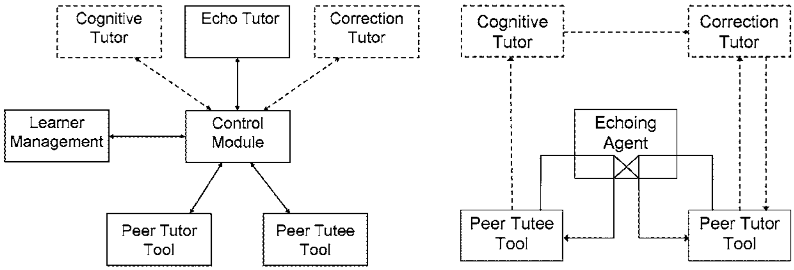


Figure 3. Changes to Cognitive Tutor Algebra architecture for peer tutoring script.

After implementing this component, we defined a new session type for "peer plus cognitive tutoring" by modifying the existing peer tutoring session type. We added two tutors to the list (the "Cognitive Tutor" and the "Correction Tutor"). Figure 3b diagrams the message passing logic within the mediator for peer plus cognitive tutoring. In this configuration, when the peer tutee takes an action, the echoing agent sends the action to the peer tutor's screen. In addition, the cognitive tutor evaluates the action, and sends the evaluation to the correction tutor. When the peer tutor takes an action, it is sent to the echo tutor, which echoes the action onto the peer tutee's screen, and to the correction tutor, which compares the peer tutor evaluation to the cognitive tutor evaluation. If responses do not match up, the correction tutor sends feedback to the peer tutor. The peer tutor can also request a hint from the correction tutor, which has stored the cognitive tutor hint for that step, and delivers it to the peer tutor.

5. Conclusions and Plans for Evaluation

We have designed and implemented a peer tutoring addition to an ITS, where students take turns tutoring each other and the ITS provides tutoring support. Our next step is to conduct a second experiment to evaluate the enhanced peer tutoring script. We will compare three conditions: the enhanced peer tutoring script (peer plus cognitive tutoring condition), the baseline peer tutoring script (peer tutoring condition), and a control where students use the ITS individually (*i*ndividual condition). We expect that students in the collaborative conditions will learn more than students in the individual conditions and students in the peer plus cognitive condition will learn the most. Learning will be measured by items from the unit, as well as transfer items that evaluate students' abilities to apply their knowledge and explain their solutions.

This work is an initial step toward combining the benefits of intelligent tutoring systems and collaborative learning activities. We have modified the Cognitive Tutor Algebra so that students can use it collaboratively to tutor each other while being provided with cognitive support by the system. Our ultimate goal is to extend our implementation to include collaborative tutoring of student interaction in order to increase deep learning.

References

[1] Koedinger, K. R., Anderson, J. R., Hadley, W. H., & Mark, M. A. (1997). Intelligent tutoring goes to school in the big city. International Journal of Artificial Intelligence in Education, 8, 30-43.
[2] Gokhale, A. (1995). Collaborative learning enhances critical thinking. Journal of Technology Education, 7(1), 22-30.
[3] Johnson, D. W. and Johnson, R. T. (1990). Cooperative learning and achievement. In S. Sharan (Ed.), Cooperative learning: Theory and research (pp. 23-37). New York: Praeger.
[4] Ritter, S., Blessing, S. B., & Hadley, W. S. (2002). SBIR Phase I Final Report 2002. Department of Education. Department of Education RFP ED: 84-305S.
[5] Kollar, I., Fischer, F., & Hesse, F. W. (in press). Collaboration scripts - a conceptual analysis. Educational Psychology Review.
[6] Harrer, A., McLaren, B. M., Walker, E., Bollen, L., and Sewall, J. (2006). Creating Cognitive Tutors for Collaborative Learning: Steps Toward Realization. User Modeling and User-Adapted Interaction: The Journal of Personalization Research (UMUAI), vol 16, nos.3-4, 175-209.
[7] Soller, A.L. (2001). Supporting Social Interaction in an Intelligent Collaborative Learning System. International Journal of Artificial Intelligence in Education, 12, 40-62.
[8] Biswas, G., Schwartz, D. L., Leelawong, K., Vye, N., & TAG-V. (2005). Learning by teaching: A new agent paradigm for educational software. Applied Artificial Intelligence, 19, 363–392.
[9] Fantuzzo, J. W., Riggio, R. E., Connelly, S., & Dimeff, L. A. (1992). Effects of reciprocal peer tutoring on mathematics and school adjustment: A componential analysis. Journal of Educational Psychology, 84, 331-339.
[10] King, A., Staffieri, A., & Adelgais, A. (1998). Mutual peer tutoring: Effects of structuring tutorial interaction to scaffold peer learning. *Journal of Educational Psychology, 90*, 134-152.
[11] Fuchs, L.S., Fuchs, D., Prentice, K., Burch, M., Hamlett, C.L., Owen, R., & Schroeter, K. (2003). Enhancing third-grade students' mathematical problem solving with self-regulated learning strategies.
[12] Webb, N.M., Troper, J.D., & Fall, R. (1995). Constructive activity and learning in collaborative small groups. Journal of Educational Psychology, 87, 406-423.
[13] Walker, E., Rummel, N., McLaren, B. M., and Koedinger. K. R. The Student Becomes the Master: Peer Tutoring in the Cognitive Tutor Algebra. To appear as a short paper to CSCL07.
[14] Walker, E., Koedinger, K. R., McLaren, B. M. and Rummel, N. (2006). Cognitive Tutors as Research Platforms: Extending an Established Tutoring System for Collaborative and Metacognitive Experimentation. Proceedings of the 8th International Conference on Intelligent Tutoring Systems.

Acknowledgment

This research is supported by the Pittsburgh Science of Learning Center, NSF Grant # 0354420.

Course-Based Experimentation

This page intentionally left blank

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Accelerated Future Learning via Explicit Instruction of a Problem Solving Strategy

Min CHI and Kurt VANLEHN
Learning Research and Development Center
University of Pittsburgh, PA, 15260

Abstract. Explicit instruction in a problem-solving strategy accelerated learning not only in the domain where it was taught but also in a second domain where it was not taught. We present data from a study in which students learned two unrelated deductive domains: probability and physics. During the probability instruction, the Strategy group was trained with an Intelligent Tutoring System (ITS) that explicitly taught a domain-independent backward chaining problem-solving strategy while the No-strategy groups trained with another ITS without any explicit strategy instruction. During the subsequent physics instruction, both groups were trained with the same ITS, which did not explicitly teach any strategy. The Strategy group gained significantly more than the No-strategy group in both domains. Moreover, their gains were evident both on multiple-principle problems, where the strategy should make problem solving more efficient, and on single-principle ones, where the strategy should make no difference. This suggests that the strategy increased the students' learning of domain principles and concepts, because that is all they had to know in order to solve the single-principle problems.

Keywords. Inter-domain transfer, Intelligent Tutoring Systems, Problem-solving Strategy

Introduction

One of the most important goals of learning is transfer, so transfer should also be at the core of AI in Education. This paper focuses on inter-domain transfer. That is, if students first learn one task domain, X, does that improve their learning of a second task domain, Y? For the remainder of this paper we will use the term "transfer" for inter-domain transfer only.

We define three types of transfer based upon how the benefits are measured in the second domain Y; *Direct Application*, *Savings*, and *Accelerated Future Learning*. For the purposes of clarity, we will use the term X students for students who studied domain X first and non-X students for those who did not.

Direct Application occurs if studying domain X increases students' competence to solving problems in domain Y. The X students receive instruction in domain X and then are tested in domain Y with no additional instruction. If they score higher on the Y-test than the non-X students, then Direct Application has occurred. For instance, Bassok and Holyoak [1] investigated transfer across two isomorphic domains, arithmetic-progression word problems in algebra and physics problems involving motion in a straight line with constant acceleration. They found that students who learned the algebra problems more likely spontaneously recognized that physics problems can be solved by the same equation. However, students who learned the physics topics exhibited little detectable transfer to the isomorphic algebra ones.

In both *Savings* and *Accelerated Future Learning*, the second domain Y is taught, not just tested. *Savings* occurs when training on X gives students a head start in learning Y even though both groups learned Y at the same rate. In Singley and Anderson's [6] study, the X and Y domains were the word processors WordPerfect and Word respectively, which share many operations and concepts. Students who had mastered WordPerfect, the X students, had a higher initial competence than novices with no prior word-processor experience, the non-X ones. Although both groups learned how to use Word at roughly the same rate, the X students had fewer elements to learn and thus took less time to master Word than the non-X ones. Both Direct Application and Savings occur when the domains share identical elements [6][7].

Accelerated Future Learning (AFL) occurs when both groups start with the same initial competence in domain Y, but the X group learns domain Y at a faster *rate* than the non-X group. Slotta and Chi [7] taught half their students, the X students, a domain-independent schema for understanding certain kinds of scientific phenomena. Then both X and non-X students studied Y, a physics text about electricity. Although the X training included no mention of electricity, the X students gained more than the non-X students when studying Y.

In some studies of transfer, the independent variable is not the presence or absence of instruction on domain X, but the different teaching method used. Schwartz et al.[5] compared inquiry methods to didactic methods for teaching domain X, and assessed the students learning in domain Y. Students who learned X via inquiry learned Y more effectively than students who learned X didactically. Unlike the other studies cited here, X and Y were different problems in the same task domain; however their study illustrates a type of AFL that is more similar to the type we observed in this study.

In this study, both X and Y are deductive domains. Like Schwartz et al.[5], we compared two different methods for teaching X. Half of the students studied X with explicit problem-solving strategy instruction while the other half studied it without such explicit instruction. Then both groups received the same instruction in the Y domain. Our primary research question is: for deductive task domains, can explicit instruction on problem solving strategies cause Accelerated Future Learning?

1. Problem solving strategies in Deductive domains

A task domain is *deductive* if solving a problem in it means producing a proof or derivation consisting of one or more inference steps each of which is the result of applying a general domain principle, operator, or rule. Deductive task domains are common parts of mathematical and scientific courses.

Broadly speaking, a *problem-solving strategy* is a policy or procedure that determines how a problem could be solved. In deductive task domains, two common strategies are *forward chaining* and *backward chaining*. Forward Chaining starts with a set of known propositions, applies a principle to some subset of them which produces at least one new known proposition. This process repeats until the problem is solved. The reasoning proceeds forwards from the known propositions toward the goal. Backward chaining is *goal-directed*. It works backward from a goal state to the known propositions by repeatedly applying deductive rules that infer the current goal from some set of subgoals. The goal is progressively broken down into subgoals until the known propositions are reached, which results in a partial solution plan. This plan is followed until more planning is necessary, and then the backward chaining is repeated.

Neither forward nor backward chaining are typically observed in a pure form in natural human problem solving [2][3] and neither one is generally taught to human problem solvers. Yet their success in guiding computer problem solvers suggests that explicitly teaching them might improve students' problem solving. Although there have been several tests of this hypothesis, they have been complicated by many problems including: confounded designs [8][9], null effects [10], teaching suboptimal strategies [4], and floor effects [11]. Despite all the work, we still lack an unequivocal answer to this simple question: given that some problem-solving strategies work so well for computer problem solving, should we teach them to human students? Moreover, would the strategy be domain-independent in that it would apply to others domains even though it was learned in just the initial one?

2. Methods

In this study, the students were divided into two groups. The Strategy group was explicitly taught the Target Variable Strategy, a domain-independent Backward Chaining problem-solving strategy [11]; the No-strategy group received no explicit instruction and was free to solve problems in any fashion. Two deductive task domains, probability and physics, were taught. Each domain contained ten major principles. In the first domain, probability, the Strategy students were required to study and use the Target Variable Strategy while the No-strategy were not required to use any specific strategy. In the second domain, physics, all students were taught in the same way and were not required to use any specific strategy.

2.1. Participants:

Data was collected over a period of four months during the fall of 2005 and early spring of 2006. We recruited 91 undergraduate students. They were required to have a basic understanding of high-school algebra, but not to have taken college-level statistics or physics courses. They were randomly assigned to the two groups. Each took from two to three weeks to complete the study over multiple sessions. Due to the winter break and length of the experiment, only 44 completed the experiment. Two students were dropped one scored perfectly on the probability pre-test and the other lacked time consistency. Of the remaining 42 participants (59.5% female), 20 were assigned to the Strategy group and 22 to the No-strategy group.

2.2. Three ITSs

Three ITSs were used in our study: Pyrenees-probability, Andes-probability, and Andes-physics. Apart from the declarative knowledge, Andes-probability and Andes-physics were identical, and we will use the term Andes to refer to them both. All three employed a multi-paned user interface that consisted of a problem statement window, a variable window, an equation window, and a dialog window. In Pyrenees, students interacted with the tutor entirely via a text-based menu and all interactions occur in the dialog window. Andes is multi-modal and the students interacted with it in all four windows using the mouse and keyboard entry.

The salient difference between Pyrenees and Andes is that Pyrenees explicitly taught the Target Variable Strategy and required students to follow it. Andes provided no explicit strategic instruction nor did it require students to follow any particular strategy. Students using Andes could input any entry, and Andes colored it green if it

was correct and red if incorrect. Students could enter an equation that was the algebraic combination of several principle applications on Andes but not on Pyrenees.

Besides providing immediate feedback, both Pyrenees and Andes provided help when asked. When an entry was incorrect, students could either fix it on their own or ask for *what's-wrong help*. When they did not know what to do next, they could ask for *next-step help*. Both Pyrenees and Andes gave similar *what's-wrong help* based upon the domain knowledge but the *next-step help* differed. Because Pyrenees required students to follow the Target Variable Strategy, it knew exactly what step the student should be doing next so it gave specific hints. In Andes, on the other hand, students could enter correct equations in any order, and an equation was considered correct if it was true, regardless of whether it was useful for solving the problem. So Andes did not attempt to figure out the student's problem-solving plans or intentions. Instead, it picked a step that it would most like to do next, and hinted that step. Both *next-step help* and *what's-wrong help* were provided via a sequence of increasingly specific hint. The last hint in the sequence, the *bottom-out hint*, told the student exactly what to do.

In this study, the Strategy students learned probability on Pyrenees and then physics on Andes-physics; while the No-strategy students learned both probability and physics on Andes.

2.3. Procedure

This study had 4 main segments: background survey, probability instruction, Andes interface training, and physics instruction (see Table 1, left column). The background survey included high school GPA, Math SAT, Verbal SAT score, and experience with algebra, probability, and physics, and other information.

Table 1: Procedure

Segment	Strategy group	No-strategy group
Background Survey	Background survey	
Probability instruction	Probability pre-training	
	Probability pre-test	
	Pyrenees video	Andes-Probability video
	Probability Training on Pyrenees	Probability Training on Andes-Prob
	Probability post-test	
Andes interface training	Andes-Probability video	
	Solve a problem with Andes-Prob.	
Physics instruction	Physics pre-training	
	Physics pre-test	
	Andes-Physics video	
	Physics Training on Andes-Physics	
	Physics Post-test	

Both the probability and physics instruction consisted of the same five phases: 1) pre-training 2) pre-test, 3) watching the video, 4) training on the ITS, and 5) post-test. During phase 1, pre-training, the students studied the domain principles. For each principle, they read a general description, reviewed some examples, and solved some single and multiple-principle problems. After solving a problem, the student's answer was marked in green if it was correct and red if incorrect, and the expert's solution was

also displayed. If the students failed to solve a single-principle problem then they were asked to solve an isomorphic one; this process repeated until they either failed three times or succeed once. The students had only one chance to solve each multiple-principle problem and were not asked to solve an isomorphic problem if their answer is incorrect. During phase 2, students took the pre-test. They were not given feedback on their answer nor were they allowed to go back to earlier questions, (this was also true of the post-tests). During phase 3, they watched a video demo of a problem solving in the corresponding ITS. During probability instruction, the Strategy students also read a text description of the Target Variable Strategy.

In phase 4, both groups solved the same probability problems or physics problems in the same order on the corresponding ITS. Each main domain principle was applied at least twice and the textbook was available for students to review. During probability instruction, the Strategy students were also able to review the textual description of the Target Variable Strategy. Finally, in phase 5 students took the post-test. In each post-test, five of the multiple-principle problems were isomorphic to training problems in phase 4. The other post-test problems (5 for probability; 8 for physics) were novel, non-isomorphic multiple-principle problems. Most of the multiple-principle problems had dead-end search paths so that the Target Variable Strategy could show an advantage in search efficiency.

Only the Strategy students took the third segment, Andes interface training. Its purpose was to familiarize them with Andes without introducing any new domain knowledge. After watching the Andes-Probability video that was given to the No-strategy students during probability instruction, the Strategy students solved a problem in Andes-probability. The problem was one of the twelve problems that they had previously solved in Pyrenees. Our pilot studies showed that solving one problem was enough for most students to become familiar with the Andes interface.

The two key procedural differences between the conditions were: 1) during the probability instruction, the Strategy trained on Pyrenees while the No-strategy trained on Andes, and 2) the Strategy students spent additional time studying the Andes interface before studying physics.

2.4. Scoring Criteria for pre-trainings, pre-tests, and post-tests

Test problems required students to derive an answer by writing and solving one or more equations. We used three scoring rubrics: *binary*, *partial credit*, and *one-point-per-principle*. Under the *binary* rubric, a solution was worth 1 point if it was completely correct or 0 if not. Under the *partial credit* rubric, each problem score was the proportion of correct principle applications evident in the solution -- a student who correctly applied 4 of 5 possible principles would get a score of 0.8. Under the One-*point-per-principle* rubric gave a point for each correct principle application. Solutions were scored by a single grader in a double-blind manner.

3. Results

Several measures showed that the incoming students' competence was balanced across conditions. There was no significant difference between groups on (1) the background survey, (2) the probability pre-test with respect to their scores on three types of problems, single-principle, multiple-principle, and overall, across all 3 scoring rubrics, or (3) the probability pre-training on all three types of problems. Thus, despite attrition, the conditions remained balanced in terms of incoming competence.

Results also showed that the two conditions did not differ on the four training times: (1) probability pre-training; (2) probability training on Pyrenees or Andes-probability, (3) physics pre-training, and (4) physics training on Andes-physics. This is fortuitous, as it implies that any difference in post-test scores is due to the effectiveness of the instruction rather than time-on-task variations.

The main outcome (dependent) variables are the students' scores on the probability post-test, the physics pre-training, pre-test, and post-test. We discuss each in turn.

On the probability post-test using the binary scoring rubric, the Strategy students scored significantly higher than the No-strategy ones: $t(40)=3.765$; $p=0.001$. The effect size was 1.17, and is denoted "d" subsequently. Moreover, they also scored higher than the No-strategy students on single-principle problems, $t(40)=3.960$; $p<0.001$; $d=1.24$, and multiple-principle ones, $t(40)=2.829$; $p=0.007$; $d=0.87$. The same pattern was found with under the partial credit and one-point-per-principle scoring rubrics.

During physics pre-training under the binary scoring rubric, the Strategy students solved more single-principle problems correctly in the *first* try than the No-strategy ones: $t(40)=2.072$, $p=0.045$; $d=0.64$. No significant difference was found between the two groups on the total number of simple- and multiple-principle problems solved correctly. This finding could be due to an unlucky random assignment: the Strategy students may have had higher prior physics knowledge than the NS. Although we cannot rule this interpretation out, there was no difference across the two conditions on the background questionnaire items relating to the students' physics experience and grades. Therefore, it is more likely that the Strategy students did, even at this early state, learn physics faster than the No-strategy students. This explanation is consistent with the results presented next.

On the physics pre-test, the Strategy students scored higher than the No-strategy under all three scoring rubrics. Under the binary scoring rubric, $t(40)=2.217$, $p=0.032$, $d=0.69$. On the single-principle problems, the two conditions did not differ significantly regardless of the scoring rubric. This was probably due to a ceiling effect. On the multiple-principle problems, the Strategy students scored higher than the No-strategy ones under the partial credit rubric, $t(40)=2.913$, $p=0.0058$, $d=0.90$ and one-point-per-principle rubric $t(40)=.800$, $p=0.008$, $d=0.86$, but not on the binary rubric $t(40)=1.148$, $p=0.147$. This could be due to the binary rubric's lack of sensitivity. In any case, the overall physics pretest scores indicate that the Strategy students learned more effectively during the physics pre-training than the No-strategy students.

On the physics post-test, the Strategy students scored much higher than the No-strategy students on the single-principle and multiple-principle problems, and overall under all three scoring rubrics. Under the binary rubric, $t(40)=4.130$, $p<0.0002$, $d=1.28$; on overall problems, $t(40)=3.211$, $p=0.003$, $d=1.00$ on single-principle problems and $t(40)=3.395$, $p<0.001$, $d=1.23$ on multiple-principle ones.

The results presented so far are consistent with two kinds of transfer: *Savings*: the Strategy students had a head start over the No-strategy ones on learning physics; or *Acceleration of Future Learning*: the Strategy students may have learned physics faster than the No-strategy ones. In order to differentiate between a head-start and a faster learning rate, we ran an ANCOVA on the physics post-test scores using the physics pre-test scores as a covariant. This yielded a post-test score for each student, adjusted for the difference in his/her physics pre-test score. Under binary scoring rubric, the Strategy students had higher adjusted post-test scores ($M=0.705$, $SD=0.21$) than the No-strategy students ($M=0.478$, $SD=0.22$). This difference was large and reliable $F(39)=11.079$, $p=0.002$, $d=1.05$. A similar pattern held for the other two rubrics. This

suggests that learning the strategy in probability did accelerate the students' learning of physics. This is intuitively satisfying, as we chose task domains that had very little overlap, making a Savings-style transfer unlikely.

The results are summarized in Table 2. Numbers in the cells are effect sizes. An *ms* or *ns* indicates that the difference between the two conditions was marginally significant or not-significant respectively. On the measures labeled ns*; the non-significance was probably due to a ceiling effect. In sum, the Strategy and No-strategy students tied on the measure of prior knowledge (the probability pre-training and pre-test), and the Strategy students scored higher than the No-strategy ones at all the subsequent assessment points: probability post-test, physics pre-test and post-test.

Table 2: Effect sizes: Strategy compared to No-strategy

scoring rubrics			Single-	Multiple-	Overall
binary	Probability	Post-test	1.24	0.87	1.17
	Physics	Pre-test	*ms*	*Ns*	0.69
		Post-test	1.00	1.23	1.28
		Adjusted Post-test	*ns**	1.02	1.05
partial credit	Probability	Post-test	1.24	1.04	1.27
	Physics	Pre-test	*ns*	0.90	0.93
		Post-test	0.92	1.17	1.21
		Adjusted Post-test	*ns**	0.78	0.81
one-point-per-principle	Probability	Post-test	1.24	0.99	1.07
	Physics	Pre-test	*ms*	0.86	0.91
		Post-test	0.67	1.17	1.18
		Adjusted Post-test	*ns**	0.73	0.75

4. Discussion

Consistent with our previous study of physics [11], the Target Variable Strategy improved students' learning significantly in the initial domain, probability. Because the Target Variable Strategy increased learning in two deductive domains, it may be an effective strategy in other deductive domains as well. More importantly, we found that teaching students the Target Variable Strategy in probability significantly accelerated their learning in a new domain, physics, even though it was not taught there. Let us consider three hypotheses about *why* teaching the Target Variable Strategy improved learning in both domains.

The first hypothesis is that teaching students the Target Variable Strategy improved their search efficiency. The multiple-principle problems were constructed so that using the strategy would reduce the average number of steps required to solve the problems, and hence reduce both time and the likelihood of error. If the search-efficiency hypothesis was true, the Strategy students should perform better than the No-strategy ones on the multiple-principle problems; while on the single-principle problems, where search is not required, the two groups should perform. However, this latter prediction did not pan out (see Table 2). The Strategy students outscored the No-strategy ones on single-principle post-test problems in both probability and physics.

The second hypothesis is that it improved students' schema learning. A schema is defined as a structure that allows problem solvers to recognize a problem as belonging to a particular category of problems that normally require particular moves. If teaching students the Target Variable Strategy improved their schema acquisition, then the Strategy students should have performed better than the No-strategy ones on the

problems that are isomorphic to training problems; on the novel, non-isomorphic test problems, however, the two groups should perform equally. This latter prediction also turned out to be false. On the non-isomorphic multiple-principle problems (5 in probability post-test and 8 in physics post-test), the Strategy students performed significant better than the No-strategy ones: $t(40)= 2.27$, $p= 0.029$ on the probability post-test and $t(40)= 3.803$, $p< 0.0005$ on the physics post-test respectively.

The third hypothesis is that teaching the strategy improves students' learning of the domain concepts and principles. Although this may seem implausible, the Target Variable Strategy requires students to apply one principle at a time (by selecting the desired principle from a menu), whereas Andes did not force them to identify the principles nor to apply them one at a time. Therefore, if students made a mistake, they would get principle-specific feedback in Pyrenees but not in Andes. Such focus on principles during probability instruction may have convinced the Strategy students to continue to focus on principles during physics instruction. If this hypothesis is true, we expect that on all types of problems, the Strategy group should perform better than the No-strategy one. This turned out to be the case. Therefore, it suggests that the main effect of teaching the strategy was to get students to focus on the domain principles in both domains.

In sum, we have shown that teaching students an explicit problem-solving strategy improved their performance in the initial domain and more importantly, it seems to cause Accelerated Future Learning in a new domain. Because the improvement occurred with all types of problems, it seems likely that the strategy instruction accelerated the learning of domain principles, as opposed to accelerating the learning of problem schemas or improving the search efficiency. It is rare to observe inter-domain transfer at all, especially one with such a large effect size. Acceleration of Future Learning is similarly uncommon as is getting the students to focus not on the whole problem solution but on each principle individually. Given that this hypothesized shift in learning orientation was obtained entirely with text and ITS, it may be possible to bring it to scale quickly, with large benefits in many deductive domains.

References

[1] Bassok M, & Holyoak K 1989. Interdomain Transfer between Isomorphic Topics in Algebra and physics. Journal of Experimental Psychology: Learning, Memory, and Cognition, 15, 1 153-166.

[2] Newell, A. and Simon, H. (1972). Human Problem Solving. Englewood Cliffs, NJ: Prentice Hall.

[3] Priest, A. G., and R. O. Lindsay (1992). "New Light On Novice-Expert Differences in Physics Problem Solving," British Journal of Psychology, 83, 389-405.

[4] Scheines, R & Sieg, W 1994.Computer environments for proof construction. Interactive Learning Environments, 4(2), 159-169.

[5] Schwartz, D., Bransford, J. et al. (2005). Efficiency and innovation in transfer. In J. Mestre (Ed.), Transfer of learning from a modern multidisciplinary perspective. CT: Information Age Publishing.

[6] Singley, K. & Anderson, R. 1989. The transfer of cognitive skill. Cambridge, MA: Harvard Press.

[7] Slotta, J. D. & Chi, M.T.H. (2006). Students Understand Challenging Topics in Science Through Ontology Training. Cognition & Instruction, 2006, Vol. 24 Issue 2, p261-289.

[8] Sweller 1988 Cognitive load during problem solving Effect on Learning. Cognitive Science12 257-285

[9] Sweller, J., & Levine, M. 1982. Effects of goal specificity on means-ends analysis. Journal of Experimental Psychology: Learning, Memory, and Cognition, 8, 463-474.

[10] Trafton, J, & Reiser, B. (1991). Providing natural representations to facilitate novices' understanding in a new domain: Forward and backward reasoning in programming. *The Proceedings of the 13th Annual Conference of the Cognitive Science Society* (pp. 923-927). Hillsdale, NJ: Erlbaum.

[11] VanLehn, K., Bhembe, D., Chi, M. et al. 2004. Implicit versus explicit learning of strategies in a non-procedural cognitive skill. 7th Conference on Intelligent Tutoring Systems, Maceio, Brazil

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Explaining Self-Explaining: A Contrast between Content and Generation

Robert G.M. HAUSMANN and Kurt VANLEHN
Learning Research and Development Center, University of Pittsburgh
3939 O'Hara Street, Pittsburgh, PA, USA

Abstract. Self-explaining has been repeatedly shown to result in positive learning outcomes for students in a wide variety of disciplines. However, there are two potential accounts for why self-explaining works. First, those who self-explain experience more content than those who do not. Second, there are differences in the activity of generating the explanations versus merely comprehending them. To compare these two accounts, the present *in vivo* classroom study, conducted in the PSLC physics LearnLab, attempted to contrast robust learning from generating explanations with the actively of studying instructional explanations. The students' learning activities (self-explaining vs. paraphrasing) were crossed with the completeness of the examples they studied (complete vs. incomplete). During a classroom period on electrodynamics, students alternated between solving problems and studying examples. On these problems, the self-explainers made fewer errors and asked for less help than paraphrasers. On homework problems done many days later, self-explainers displayed evidence of far transfer in a related, yet new domain (i.e., magnetism). In conclusion, prompting students to self-explain while studying examples, in an authentic classroom environment, can result in positive near- and long-term learning.

Keywords. self-explaining, paraphrasing, study strategies

1. Introduction

When students study instructional materials, including textbooks, worked-out examples, diagrams, and other multimedia materials, they may *self-explain* it, which means generating an explanation for what they have just read based on their knowledge or on content read earlier [1]. While self-explaining has consistently been shown to be effective in producing robust learning gains in several domains, including physics [2], the human circulatory system [3; 4], and algebra and geometry [5], the underlying sources of the effect are still being investigated. Identifying these sources is an important research objective because the design of effective educational practices and software can be guided by a deeper understanding of the process and outcomes of self-explaining. The present study attempts to add to this understanding, especially as it occurs in a real-world context: the classroom.

2. Explaining self-explaining

The first studies of self-explanation, which were based on analyses of verbal protocols, showed that the amount of self-explaining correlated strongly with performance on post-test measures of problem-solving performance [2; 6; 7]. Subsequent studies showed that students who were prompted to self-explain sentences in a scientific text learned more than students who were asked to paraphrase the sentences instead [2; 8; 9]. Other studies showed that self-explaining could be the elicited by computers [5; 9; 10; 11].

Because these studies compared self-explanation to the lack of any explanation at all, it is not clear why self-explanation produced learning. On the one hand, self-explaining generates additional information, namely the explanations themselves, that are not present in the instructional materials. Perhaps if students were simply given these explanations, they would learn just as much [12]. Alternatively, learning from self-explaining might arise from the activity of producing the explanations. Thus, if they were given the explanations, they would *not* learn just as much. In other words, is it merely *attending* to the explanations that matters, or is robust learning more likely to occur if students *generate* the explanations themselves? Let us label these hypotheses as follows:

1. *Attention*: learning from self-generated explanations should produce comparable learning gains as author-provided explanations, provided the learner pays attention to them. Both self-generated and author-provided explanations should exhibit better learning than no explanation.
2. *Generation*: learning from self-generated explanations should produce greater learning gains than author-provided explanations because they are produced from the students' own background knowledge; however, author-provided explanations should be comparable to no explanation.

There have only been a few empirical studies that attempt to separate the Attention hypothesis from the Generation hypothesis [13]. An exemplary case can be found in a study by Lovett [14] in the domain of permutation and combination problems. Lovett crossed the source of the solution (subject vs. experimenter) with the source of the explanation for the solution (subject vs. experimenter). For our purposes, only two of the experimental conditions matter: (1) the experimenter-subject condition, wherein the students self-explained an author's solution, and (2) the experimenter-experimenter condition, wherein the students self-explained an author's solution, whereas in the experimenter-experimenter condition, the students studied an author-provided explanation. Lovett found that the experimenter-experimenter condition demonstrated better performance, especially on far-transfer items. Lovett's interpretation was that the experimenter-experimenter condition was effective because it contained higher quality explanations than those generated by students. Consistent with this interpretation, when Lovett analyzed the protocol data, she found that the participants who generated the key inferences had the same learning gains as participants who read the corresponding inferences. Thus, of our two hypotheses, Lovett's experiment supports the Attention hypothesis: the content of self-explanations matters, while the source of the explanation does not.

Brown and Kane [15] found that children's explanations, generated either spontaneously or in response to prompting, were much more effective at promoting transfer than those provided by the experimenter. In particular, students were first told a story about mimicry. Some students were then told, "Some animals try to look like a scary animal so they won't get eaten." Other students were asked first, "Why would a furry caterpillar want to look like a snake?" and if that did not elicit an explanation, they were asked, "What could the furry caterpillar do to stop the big birds from eating him?" Most students got the question right, and if they did, 85% were able to answer a similar question about two new stories. If they were told the rule, then only 45% were able to answer a similar question about the new stories. This result is consistent with the Generation hypothesis, which is that an explanation is effective when the student generates it. However, the students who were told the rule may not have paid much attention to it, according to Brown and Kane.

In summary, one study's results are consistent with the Attention hypothesis, and the other study's results are consistent with the Generation hypothesis, but both studies confounded two variables. In the Lovett study, the student-produced and author-provided explanations were of different qualities. In the Brown and Kane study, the students in the author-provided explanations condition may not have paid much attention to the explanations.

To contrast the Generation and Attention hypotheses, we conducted an experiment in the physics LearnLab (www.learnlab.org), which is a course that is designed to conduct rigorous, *in vivo* experiments on issues related to robust learning. Robust learning is defined in three parts. First, learning is considered robust when the knowledge is retained over a significant period of time. Second, robust learning is the opposite of inert knowledge in the sense that students are able to broadly apply their knowledge to other problems within the same class. Finally, robust learning can be applied to a different, yet related, domain. Robust learning is contrasted with "normal" learning, which is the performance on problems that are near transfer and retained over a relatively short duration.

3. Materials

The training materials[1] used for the present experiment were developed in association with one of the LearnLab instructors and two other physicists. The experiment covered the domain of electrodynamics, with an emphasis on the forces acting on a charged particle due to the presence of an electric field. The homework problems were near- and far-transfer versions of the training problems, along with problems from the magnetism chapter. Magnetism was used as a test-bed for studying transfer across domains. The data were collected in the LearnLab, as opposed to the laboratory, because the realism of the classroom increases the generalizability of the results without sacrificing randomization or control over extraneous variables.

3.1. Design

The experiment was a 2 x 2 between-subjects design, which crossed two independent variables: Study Strategy (paraphrasing vs. self-explaining) and Example Type

[1] For a copy of the experimental materials, visit: http://andes3.lrdc.pitt.edu/~bob/mat/exper1.html

(complete vs. incomplete). When students studied a complete example, they were given an author-provided explanation, and both the paraphrasing (PP) and self-explanation (SE) instructions were intended to insure that they paid attention to it. When students studied an incomplete example, the self-explanation prompts were intended to elicit self-explanation, while the paraphrase instructions were intended to block it. The instructions were modeled after earlier studies [2; 9], where verbal protocols confirmed that they had the intended effects of eliciting or suppressing self-explanations. Students were block randomized into one of the following four experimental conditions: paraphrase complete examples ($n = 26$), paraphrase incomplete examples ($n = 23$), self-explain complete examples ($n = 27$), and self-explain incomplete examples ($n = 28$). The two hypotheses make the following predictions:

- *Attention*: SE-complete = SE-incomplete = PP-complete > PP-incomplete
- *Generation*: SE-complete = SE-incomplete > PP-complete = PP-incomplete

4. Procedure

One hundred and four students, recruited from five sections of a second-semester, calculus-based physics course taught at the U.S. Naval Academy, were given course credit for their participation. The experiment took place in one of the open class periods, which were approximately 110 minutes in duration.

The students were introduced to the experiment and shown the instructions. Then students were prompted to solve the first problem, which was a warm-up problem to get the students acquainted with the Andes interface. Andes (www.andes.pitt.edu) is an intelligent tutoring system, created for helping students solve homework problems from the first two semesters of physics [16]. After solving the first problem, the students then studied the first example. This process, alternating between solving problems and studying examples [17], repeated for three cycles so that by the end of the training, four problems were solved and three examples were studied.

While the students were studying the examples, they were prompted to either paraphrase or self-explain at the end of each segment. To capture their verbalizations, each student wore a pair of headphones equipped with a close-talk, noise-cancelling microphone. In addition to audio, all of the on-screen activity was recorded using a screen-logging facility built into the Andes interface.

The following data-streams were created for each student: 1) an audio track of their verbalizations; 2) a video of their on-screen activities; and 3) a text-only log file of each action taken in the Andes interface. In addition to the in-class experimental session, log files from the assigned Andes homework problems were made available to the researchers. Problems from the electric and magnetic fields chapters were of particular relevance. The course instructor also provided the paper-and-pencil chapter exam that included a problem with a charged particle moving in an electric field.

5. Results

This section is broken down into three sub-sections, corresponding to the type of learning that was measured: normal and robust, which included within- and between-

domain, far-transfer items. Normal learning was defined as problem-solving performance on the items administered on the day of the experiment. Robust within-domain, far-transfer learning took place separately from the experiment, while the students solved either homework (Andes) or exam (paper-and-pencil) problems. Robust between-domain, far-transfer learning was performance on Andes homework problems, taken from a topic that was separate from the experimental materials (i.e., magnetism).

Because Andes insured that students always found a correct solution to the problem, our main dependent measure was their normalized assistance score on the problem, which was defined as the sum of all the errors and requests for help on that problem divided by the number of entries made in solving that problem. Thus, lower assistance scores indicate that the student derived a solution while making fewer mistakes and getting less help, and thus demonstrating better performance and understanding.

5.1. Normal learning: Problems solved during the experimental session

To ensure equivalent experimental conditions, we analyzed performance on the warm-up problem, which revealed no reliable differences, $F(3, 99) = .37$, $p = .78$. In terms of normal learning, normalized assistance scores were contrasted by averaging over individuals for all three problems in the training set. Using a repeated-measures Analysis of Variance (ANOVA), a main effect for Study Strategy was observed, with the self-explanation condition demonstrating lower normalized assistance scores than the paraphrase condition (see Figure 1), $F(1, 73) = 6.19$, $p = .02$, $\eta_p^2 = .08$. This result is consistent with the Generation hypothesis.

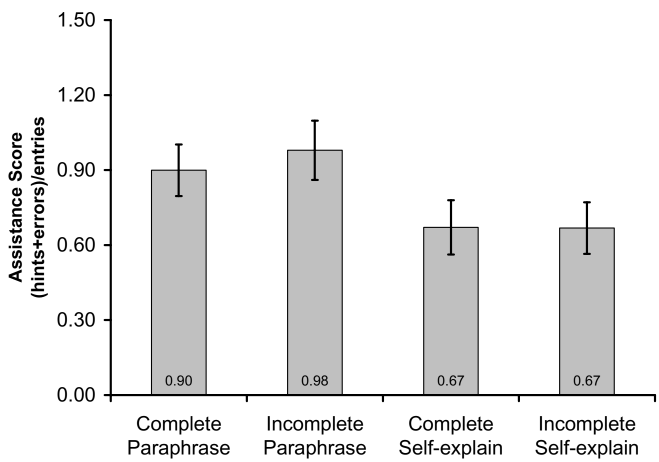


Figure 1. Means and standard errors on the three training problems.

5.2. Robust learning: Within-domain, far transfer

On the chapter exam, administered 29 days after the experiment, there were neither reliable main effects nor an interaction; however, post-hoc comparisons using the

Fischer LSD test revealed that the complete self-explanation condition (M = 90.83, SD = 9.96) had a marginally higher score than the complete paraphrase condition (M = 73.00, SD = 22.51) (LSD, p = .06). The reason why a difference was not observed could be due to the insensitivity of the measure or that the students' preparation for the exam washed out any potential differences between conditions.

To overcome these potential limitations, we analyzed the students' performance on a homework problem that was isomorphic to the chapter exam, in the sense that they shared an identical deep structure (i.e., both analyzed the motion of a charged particle moving in two dimensions). The homework problem was solved after the training but before the chapter exam. The marginal effect observed on the chapter exam was replicated on the isomorphic homework problem. There was a reliable main effect for Study Strategy, with the self-explanation condition demonstrating lower normalized assistance scores than the paraphrase condition, $F(1, 27) = 4.07$, $p = .05$, $\eta_p^2 = .13$ (see Figure 2). This result is also consistent with the Generation hypothesis.

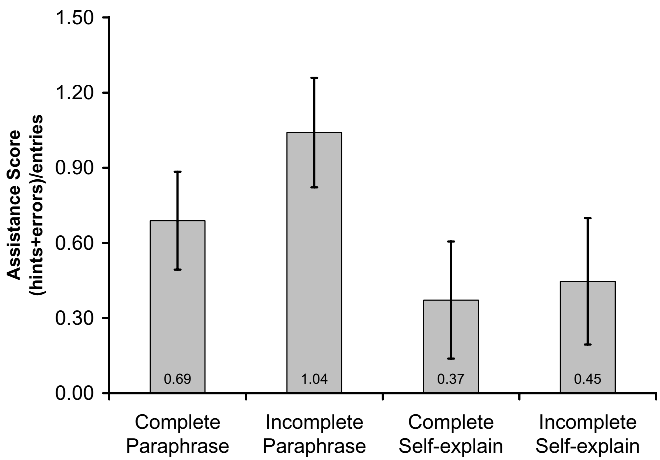


Figure 2. Means and standard errors on a homework electric-field problem.

5.3. Robust learning: Between-domain, far transfer

Another component of robust learning is to examine performance on problems in a different domain. To measure far transfer between domains, we analyzed homework performance in the magnetism chapter. We chose a magnetism problem that was somewhat similar to the electrodynamics training problems, although there remained major differences due to the differences in the task domain. The similarities were that both the principle knowledge component from electrodynamics (i.e., the definition of an electric field: $\mathbf{F_e} = q\mathbf{E}$) and magnetism (i.e., the vector expression for a magnetic force on a charged particle moving in a magnetic field: $\mathbf{F_b} = q\mathbf{v}\times\mathbf{B}$) were vector equations. Second, both problems included an opposing force that was equal in magnitude, but opposite in direction; thus, there was no acceleration in either case.

Using an ANOVA, a main effect for Study Strategy was observed, with the self-explanation condition demonstrating lower normalized assistance scores than the

paraphrase condition (see Figure 3), $F(1, 46) = 5.22$, $p = .03$, $\eta_p^2 = .10$. Once again, the results favor the Generation hypothesis.

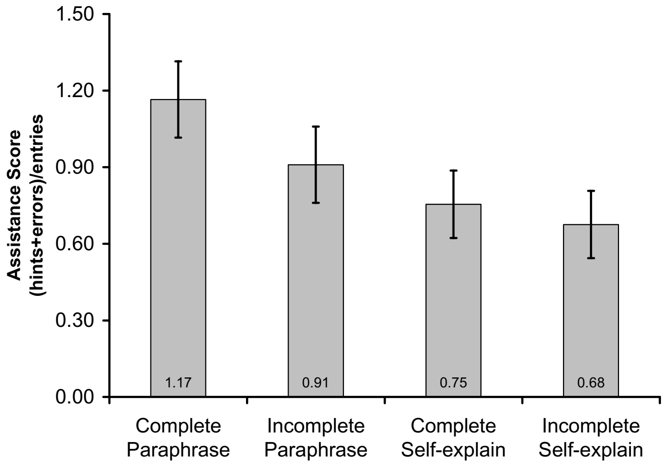


Figure 3. Means and standard errors on a homework magnetic-field problem.

6. Conclusions

Prior to this study, it appeared that the self-explanation effect could be due simply to attending to the content of the explanations, and that it did not matter whether the explanations were provided by an author or produced by the student. This hypothesis, which we called the Attention hypothesis, was not supported by our data. When students paraphrased high-quality explanations, we found less learning than students who were prompted to generate their own explanations of examples which omitted the author-provided explanations. Paraphrasing was used so that we could be reasonably sure that they at least paid some attention to the explanations. Thus, the evidence favors the Generation hypothesis, which asserts that the important variable for learning was the *process* of producing an explanation. In future work, we hope to gain insights into the underlying process by analyzing the recordings collected during the study.

This study also demonstrates that a fairly robust finding from the laboratory [2] could be imported into the classroom. The evidence suggests that prompting for self-explaining can be replicated in the real world, even when the conditions are such that the students are prompted to self-explain worked-out video examples in a noisy, complex environment such as a classroom. Prompting for self-explaining had a positive influence on normal learning. Normal learning consisted of a short retention interval, and problems that were designed to be isomorphic to the worked-out examples. The positive influence of self-explaining on learning has been shown on normal learning in previous laboratory studies [4]; however, the present study takes this result one step further by evaluating the effect on robust learning measures, including features such as long retention intervals, far transfer within and between domains.

7. Acknowledgements

This project was funded by the Pittsburgh Science of Learning Center under grant 0354420 from the NSF. The authors would like to thank the anonymous reviewers for their helpful comments; Robert Shelby and Brett van de Sande for their assistance in material development; Michael Ringenberg, Min Chi, and Anders Weinstein for their technical expertise; and we are extremely grateful to Donald Treacy, John Fontanella, and Mary Wintersgill for graciously allowing us to collect data in their classrooms.

8. References

[1] Roy, M., & Chi, M. T. H. (2005). The self-explanation principle in multimedia learning. In R. E. Mayer (Ed.), The Cambridge handbook of multimedia learning (pp. 271-286). Cambridge: Cambridge University Press.
[2] Chi, M. T. H., Bassok, M., Lewis, M. W., Reimann, P., & Glaser, R. (1989). Self-explanations: How students study and use examples in learning to solve problems. Cognitive Science, 13, 145-182.
[3] Ainsworth, S., & Loizou, A. T. (2003). The effects of self-explaining when learning with text or diagrams. Cognitive Science, 27(4), 669-681.
[4] Chi, M. T. H., DeLeeuw, N., Chiu, M.-H., & LaVancher, C. (1994). Eliciting self-explanations improves understanding. Cognitive Science, 18, 439-477.
[5] Aleven, V. A. W. M. M., & Koedinger, K. R. (2002). An effective metacognitive strategy: Learning by doing and explain with a computer-based Cognitive Tutor. Cognitive Science, 26, 147-179.
[6] Renkl, A. (1997). Learning from worked-out examples: A study on individual differences. Cognitive Science, 21(1), 1-29.
[7] Bielaczyc, K., Pirolli, P., & Brown, A. L. (1995). Training in self-explanation and self-regulation strategies: Investigating the effects of knowledge acquisition activities on problem solving. Cognition and Instruction, 13(2), 221-252.
[8] Magliano, J. P., Wiemer-Hastings, K., Millis, K. K., Munoz, B. D., & McNamara, D. S. (2002). Using latent semantic analysis to assess reader strategies. Behavior Research Methods, Instruments & Computers, 34(2), 181-188.
[9] Hausmann, R. G. M., & Chi, M. T. H. (2002). Can a computer interface support self-explaining? Cognitive Technology, 7(1), 4-14.
[10] Conati, C., & VanLehn, K. (2000). Toward computer-based support of meta-cognitive skills: A computational framework to coach self-explanation. International Journal of Artificial Intelligence in Education, 11, 398-415.
[11] McNamara, D. S., Levinstein, I. B., & Boonthum, C. (2004). iSTART: Interactive strategy training for active reading and thinking. Behavioral Research Methods, Instruments, and Computers, 36, 222-233.
[12] Cavalli-Sforza, V. (1999). Constructed vs. received representations for scientific argument: Implications for learning and coaching. In M. Hahn & S. C. Stoness (Eds.), Proceedings of the Fourteenth Annual Conference of the Cognitive Science Society. Mahwah, NJ: Erlbaum.
[13] Schworm, S., & Renkl, A. (2002). Learning by solved example problems: Instructional explanations reduce self-explanation activity. In W. D. Gray & C. D. Schunn (Eds.), Proceedings of the 24th Annual Conference of the Cognitive Science Society (pp. 816-821). Mahwah, NJ: Erlbaum.
[14] Lovett, M. C. (1992). Learning by problem solving versus by examples: The benefits of generating and receiving information. In Proceedings of the Fourteenth Annual Conference of the Cognitive Science Society (pp. 956-961). Hillsdale, NJ: Erlbaum.
[15] Brown, A. L., & Kane, M. J. (1988). Preschool children can learn to transfer: Learning to learn and learning from example. Cognitive Psychology, 20(4), 493-523.
[16] VanLehn, K., Lynch, C., Schultz, K., Shapiro, J. A., Shelby, R., Taylor, L., et al. (2005). The Andes physics tutoring system: Lessons learned. International Journal of Artificial Intelligence and Education, 15(3), 147-204.
[17] Trafton, J. G., & Reiser, B. J. (1993). The contributions of studying examples and solving problems to skill acquisition. In Proceedings of the Fifteenth Annual Conference of the Cognitive Science Society (pp. 1017-1022). Hillsdale, NJ: Erlbaum.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Out of the Lab and into the Classroom: An Evaluation of Reflective Dialogue in Andes

Sandra KATZ, John CONNELLY, and Christine WILSON
Learning Research and Development Center, University of Pittsburgh, Pittsburgh, PA 15260

Abstract. Several laboratory studies have demonstrated the effectiveness of presenting students with "Reflection Questions" immediately after they have solved quantitative problems [1, 2]. The main goal of the two experiments described in this paper was to determine if the positive results of post-practice reflection from laboratory studies hold up in an actual classroom setting. We added prototype reflective dialogues to the Andes physics tutoring system [3, 4] and evaluated them in a course at the US Naval Academy. Consistent with prior research, the dialogues promoted conceptual understanding of physics. However, measures of retention and transfer to problem-solving ability showed only a marginal effect of completing the reflective dialogues, in the first experiment.

Keywords. Reflection, classroom-based research, natural-language interaction

Introduction

A well-known problem in the teaching of quantitative subjects such as physics is that even high-achieving students often leave a course with shallow knowledge [5, 6]. One technique that human tutors use to support students in tying a problem with its associated concepts, and in formulating abstract solution schemata from a set of similar problems, is to ask them qualitative questions immediately after they have solved a problem and to guide them in answering these questions. Such questions are commonly called "reflection questions" (RQs) [2], and the student-tutor discussions that stem from them are known as "post-practice reflective dialogues" (PPRs) [1].

Several studies which were conducted in a laboratory setting have shown that reflection questions and their ensuing PPRs support the development of conceptual knowledge and problem-solving ability [1, 2]. The main goal of the two "*in vivo*" experiments described in this paper was to determine if these results would hold up in an actual classroom setting. Toward this end, we developed prototype reflective activities for the Andes physics tutoring system [3, 4] and evaluated them in a physics course at the US Naval Academy.

Andes seemed like an ideal environment in which to couch our manipulation. Although Andes has been shown to significantly enhance students' problem-solving performance when compared with students in "traditional" sections of the course that do homework using paper-and-pencil, evaluations have not shown conclusively that the tutor enhances conceptual knowledge [4]. Furthermore, several physics instructors commented that they would like the tutor to do more to combat "shallow learning."

1. Overview of Andes

Andes is described in detail in [3] and summarized in [4]. We review Andes' essential features as background for our discussion of the reflective activities that we added to it.

Andes was designed to be used with most textbooks for first-year college physics courses, or for advanced high school courses. It provides over 500 problems for students to solve, just as they would with pencil and paper, but with the added benefit of coaching when they get stuck and immediate feedback on the correctness of their actions. Andes' coaching is designed to be increasingly directive, with general hints followed by more specific advice about what to do next and how to do it. Conceptual knowledge is mainly included in the top-level hints. However, many students gloss over these hints in order to get to the most detailed "bottom out" hints, which can lead to shallow learning [5]. We expected that students would be more receptive to instruction about physics concepts and problem-solving schemata *after* they solved problems than during problem solving, when they are intent on getting the correct answer. Andes is available for download at www.andes.pitt.edu.

2. Experiment 1

2.1 Background and Motivation

In addition to determining whether PPRs would support learning *in vivo*, the first experiment tested our prediction that the more *interactive* students are when responding to reflection questions, the more they will learn. This hypothesis is grounded in constructivist learning theory, as well as in empirical research that suggests that interaction—mutual contributions to dialogue by the student and the tutor—is a critical factor in the effectiveness of one-on-one human tutoring [7].

We compared three versions of reflective activity at different levels of interactivity. In each reflection condition, RQs were presented to students sequentially in a pop-up window as soon as they entered the correct solution to an Andes problem. This window was separate from the Andes interface, which remained visible in the background as students worked on the RQs.

Table 1 shows an RQ that was asked after a work-energy problem and an excerpt from the ensuing dialogue. "What if" questions such as this one were used throughout the intervention. In the least interactive form of reflection, the student entered a response to an RQ and then was shown an expert-generated response as feedback. The tutor then went on to the next question. We call this the canned-text response condition (CTR). It was designed to stimulate self-explanation, but not interaction.

The second reflection condition used Knowledge Construction Dialogues (KCDs) to guide students to a correct answer to each reflection question. KCDs are designed to guide students through Socratic-style questioning in learning one or a small number of concepts [8]. When a student enters an incorrect response to one of the tutor's questions, the dialogue engine invokes a remedial sub-dialogue. Table 1 shows a sample KCD (T = tutor; S = student). Lines 3-4 illustrate remediation. We refer to this form of reflection as the KCD condition.

Table 1. A sample reflective dialogue

Andes Problem: A woman pulls a suitcase with a strap which makes an angle of 37.0 deg with the horizontal. The suitcase has a mass of 15.0 kg, and has wheels on it so that it can be considered to move without friction. If the tension in the strap is 80.0 N, what work is done on the suitcase by the strap when it is displaced 2.50 m horizontally?	
Reflection Question: Suppose that the suitcase has no wheels, and is instead subject to a frictional force as it slides across the floor. If everything else were the same, how would the work done by the strap on the suitcase change?	
1.	**T:** What quantities do you need to calculate work?
2.	**S:** force, displacement
3.	**T:** Unless you know both the force and displacement **vectors**, you also need the **angle** between the force and the displacement to calculate work.
4.	**T:** In short, to calculate work you need the force vector and the displacement vector or, alternatively, the magnitude of the force, the magnitude of the displacement, and the angle between the force and the displacement. What type of force does the strap exert on the suitcase?
5.	**S:** tension
6.	**T:** Good. Does the tension force have a component in the direction of the suitcase's displacement?...

Although the KCD version of reflection is more interactive than the canned text version, it nonetheless does not support mixed initiative. For example, the student cannot ask a question. Our third reflection condition implemented a modest form of mixed initiative. At various points in the dialogue there were hyperlinks following the tutor's turn, which were associated with a question that a student might want to ask at that point in the dialogue. If the student selected one of these questions, the tutor answered the question and then returned to the main line of reasoning. We refer to this more interactive form of reflection as the mixed-initiative condition (MIX).

Based on the interactivity hypothesis, we predicted that students in the MIX condition would outperform students in the standard KCD condition, who would in turn outperform students in the CTR condition. In addition, we expected that students in any reflective condition would outperform students in a control condition (Ctrl) who used standard Andes without reflective dialogues.

2.2 Method

Our intervention consisted of 22 reflection questions, distributed across 9 problems in the work-energy (WE) unit of a first-year physics course at the US Naval Academy. The experiment was conducted in the fall of 2005 and lasted for about three weeks.

The experiment used a pre-test→intervention→post-test design. Both tests were administered to students in a laboratory setting. However, students did the intervention as homework, whenever they wished. The two tests were identical in form and content, with 15 multiple choice questions (12 qualitative and 3 quantitative WE problems).

Participants were 123 midshipmen (94 men, 22 women), enrolled in seven sections of General Physics I and taught by four instructors. They were randomly assigned to conditions within each course section. Conditions were balanced within category of academic major (Humanities, Engineering, Math/Science) and by sorted QPA within major category. There were 30-32 students per condition.

In order to measure retention and transfer, we analyzed copies of the hourly exam that followed the work-energy unit and of the final exam. The final exam was comprehensive; it contained seven work-energy problems. Both tests were strictly quantitative, thus allowing us to determine if there was a difference in the degree to which each condition supported problem-solving ability.

2.3 Results and Discussion

First, the bad news: Student participation in both homework problem solving and reflection was uneven across conditions and too low overall to allow us to reliably determine whether more interactive forms of reflection support learning better than less interactive forms. Among the 93 students assigned to one of the three treatment conditions, only 13 (14%) completed all 22 RQs prior to the post-test; 42 (45%) completed none of the RQs. The mean number of RQs completed was only 6.7.

Although it is possible that low participation of treatment subjects was due to dissatisfaction with the reflective dialogues, we believe that a more plausible explanation is that there was no course incentive (e.g., grade) for doing the dialogues. Many students did not do their homework, so they did not encounter the dialogues. Although instructors encouraged students to do the dialogues, they were not yet confident enough in the intervention to require students to do them.

Before doing a between-group comparison of student performance, we omitted data from students with a post-test duration of less than four minutes and a prevalence of "don't know" selections. These students obviously were not trying to do well on the post-test. We also re-classified treatment subjects who did no dialogues as control subjects and combined the KCD and MIX conditions into one "dialogue" condition because very few MIX subjects asked follow-up questions. After these regroupings, we had 48 Ctrl, 17 CTR, and 38 dialogue subjects. Given the low participation in reflection across conditions, it is not surprising that we found no significant differences in post-test score or gain score—neither by ANCOVA nor by linear regression with major, Quality Point Average (QPA), and pre-test score as covariates.

Now for some good news: We were able to determine that having students engage in some form of reflective dialogue is better than no reflective activity. Using a yoked pairs analysis, we compared "treated" with "untreated" subjects—that is, students who responded to at least 5 RQs, in any reflection condition, with students who saw no RQs. We yoked subjects by pre-test scores and major, resulting in 19 pairs. As predicted, this analysis revealed that the mean gain score was significantly higher for treated than untreated subjects; $M = 1.53$, $t(18) = 2.40$, $p = .03$. Consistent with this result, a regression analysis treating number of RQs done as a continuous variable showed that this factor significantly predicted post-test score, as opposed to the number of problems that students completed; $R^2 = .35$, $F(5,97) = 10.33$, $p < .0001$.

The results of retention and transfer analyses were mixed. Using the same re-classification of subjects discussed previously, we found that students in the canned text condition did marginally better than students in the other three conditions on the seven work-energy problems on the final exam; $F(2, 119) = 2.74$, $p = .07$. For the hourly exams, we had data from only two course sections. Both exams contained only two work-energy problems. On one exam, regressing on the number of problems that students solved prior to the exam and the number of RQs that they completed showed a significant positive effect of the latter ($p = .03$), but only a marginal overall regression.

On the other exam, the same regression analysis showed a significant positive effect for the total number of problems done, but not for the number of RQs done. Again, the overall regression was marginal.

3. Experiment 2

Our second *in vivo* experiment took place a year later, in the fall of 2006. Our first goal was to verify the main result of the first experiment, which was that students who engaged in reflective activity after solving Andes problems outperformed control subjects who did not, as measured by post-test scores and gain scores over the pre-test. Secondly, we wanted to develop and test a reflective intervention that would enhance students' problem-solving ability, in addition to their conceptual knowledge.

A positive outcome of the first experiment was that it increased instructors' confidence in our intervention. Consequently, they requested a "Reflective Follow-up" module that covered most course units (not just work-energy), provided us with feedback on the reflective dialogues throughout the development process, and required treatment subjects to complete the dialogues. We used several methods to enforce participation, which we describe in the next section.

Our design of a new set of reflective dialogues that could potentially increase students' conceptual knowledge and problem-solving ability was motivated by research done by Dufresne, Gerace, Hardiman, & Mestre [9], who distinguish between three components of a problem-solving strategy: *what* a problem's main quantities, concepts and/or principles are; *how* to map principle applications to relevant equations, manipulate equations to solve for desired quantities, and extend manipulations into related contexts; and *why (not)* to apply a given principle or solve an alternate scenario via a different approach. Dufresne et al. [9] developed and tested a solution-planning tool (HAT, for *Hierarchical Planning Tool*) that guided students through each component. Although they found some improvements in problem-solving ability, they also observed that even high-achieving students were often unable to derive a correct set of equations using the tool. We speculated that HAT planning dialogues covered too much material, because they targeted all three knowledge components.

In light of these observations, we developed post-practice reflective dialogues that targeted mainly (but not solely) one component at a time and in order, within selected units of the course. The earliest assigned problem(s) in each unit contained a *what* reflective dialogue, middle problem(s) contained a *how* dialogue, and later problem(s) contained a *why* dialogue. We then capped off each unit with a pre-solution planning dialogue that integrated these components, as did HAT. We refer to these planning dialogues as "capstone dialogues."

This experiment compared students in two conditions. The treatment condition engaged in *what*|*how*|*why* reflective dialogues and capstone dialogues, implemented using standard, tutor-led KCDs. The control condition used standard Andes without KCDs, but solved more problems than did treatment subjects to control for time on task. We did not attempt to compare different versions of reflection in this experiment, as we did in the first experiment, partly because only three sections of the course participated (vs. seven sections during the previous fall), and we wanted to optimize the power of our analyses. Instead, we compared direct, explicit instruction in

conceptual knowledge and solution planning with implicit instruction—that is, knowledge acquired through practice (via solution of extra homework problems).

3.1 Method

We again conducted this experiment *in vivo*—that is, within first-year physics classrooms at the US Naval Academy. Three sections of General Physics I, taught by two instructors, participated. Participants were 67 midshipmen (10 women, 57 men) who were randomly assigned to conditions within each course section. There were 33 treatment subjects and 34 control subjects. As before, conditions were balanced by academic major and by sorted QPA within each major category.

Our reflective intervention consisted of 21 post-practice dialogues, which targeted the *what*, *how*, and *why* components of problem solving in succession within selected course units. The intervention spanned five units (statics; translational dynamics, including circular motion; work-energy; power; and linear momentum, including impulse) lasting approximately eight weeks. In addition, there were five capstone planning dialogues (one toward the end of each unit). Control subjects solved five more problems than did treatment subjects.

As before, the experiment used a pre-test→intervention→post-test design and both tests were administered in a laboratory setting. Students solved the Andes problems as homework, on their own time. The two tests were identical in form and content, with 30 multiple choice questions (24 qualitative and 6 quantitative). To measure retention and transfer to problem-solving ability, we analyzed an hourly exam that covered several units, and the final exam. Both tests consisted solely of quantitative problems.

We implemented several strategies to ensure greater student participation in this experiment, relative to that of the first. With the course instructors' approval, Andes was modified so that students were required to do the target problems in a fixed sequence. (Ordinarily, students can do any problem in any order.) Students could not work on a problem until they solved all of its prerequisite problems and—if they were in the treatment condition—until they completed all of the dialogues associated with these problems. In addition, the KCDs were designed to discourage null responses. If students did not answer, the tutor prompted them to do so. We also tried to improve system recognition of student input by restating selected questions that a student answered incorrectly, giving the student another chance to answer via a multiple choice menu of correct and incorrect responses.

3.2 Results and Discussion

Analyses of pre- and post-test scores were more encouraging than for Experiment 1. After omitting scores from one student with a post-test duration of less than two minutes, we were left with treatment and control groups of equal size (n = 33). We also re-classified two treatment subjects who did no dialogues as control subjects (effective treatment and control ns = 31 and 35, respectively), although these subjects were not able to access the five extra control-group Andes problems. As we had hoped, ANOVA showed no significant differences between the effective treatment and control groups on pre-test score (respective Ms = 12.10 and 11.77; $F < 1$), but treatment subjects had higher mean post-test scores (17.97 vs. 15.57; $F(1, 64) = 4.89$, $p = .031$), mean raw gain scores (5.87 vs. 3.80; $F = 5.62$, $p = .021$), and mean Estes gain

scores (0.330 vs. 0.208; $F = 6.74$, $p = .012$) than did control subjects. In short, without regard to problem-solving practice, subjects who did KCDs did significantly better on the post-test than those who did not.

Student participation in both homework problem solving and dialogues was much improved over Experiment 1, with 21 of 34 control subjects (62%) and 23 of 33 treatment subjects (70%) finishing at least 80% of the assigned target problems (of 31 for control, 26 for treatment) prior to the post-test and with 25 treatment subjects (76%) finishing at least 80% of the 26 associated KCDs. However, participation was still far from perfect; 7 treatment subjects (21%) completed half or fewer of the assigned KCDs on time, and 2 of them completed no target problems or KCDs on time. We again treated KCD completion and target problem completion as continuous independent variables in regression analyses. Regressing post-test score on pre-test score, QPA, number of KCDs completed, and number of target problems completed ($R^2 = .52$, $F(4, 61) = 16.70$, $p < .00001$) showed positive contributions of all factors, but only pre-test score, QPA, and KCD completion were statistically significant (*p*s < .001, .05, & .05, respectively); problem completion was *ns* ($p = .54$). Therefore, across all subjects it was KCD completion, as opposed to target homework problem completion, that significantly predicted post-test performance.

To measure retention and transfer, we also analyzed scores from an hourly exam administered by one instructor to his two sections (n=47). All problems on this exam were quantitative and covered a subset of the units targeted during the intervention. Scores on this exam ranged from 176 to 382 (of 400), and were highly correlated with pre- and post-test scores; r*s*(45) = .54 and .71, *p*s < .0001 and .00001. However, ANOVAs showed no differences between groups (*F*s < 1), and regression of subscores on QPA, KCDs completed, and target problems completed ($R^2 = .54$, $F(3, 43) = 16.58$, $p < .00001$) showed positive contributions of all factors but a significant effect of only QPA ($p < .00001$). Therefore, student performance on the hourly exam was not significantly affected by either KCD completion or target problem completion.

In summary, student performance on the post-test relative to the pre-test was significantly influenced by the number of dialogues they completed, as opposed to the number of target problems they completed prior to the post-test. However, neither measure had a significant effect on student performance on an exam that focused on problem-solving ability. Analyses of the final exam data are in progress.

4. Conclusion

The two studies described in this paper suggest that the positive results of reflection from laboratory studies hold up in an actual classroom setting. Although our first experiment did not allow us to test our hypothesis that more interactive forms of reflection are more beneficial than less interactive forms, such as canned text, we found some evidence against this hypothesis. For example, in the first experiment, the canned text condition marginally outperformed the other three conditions on the final exam. Several studies comparing human or automated dialogue with canned text have also failed to show an advantage of the former [1, 8].

Analyses of retention and transfer to problem-solving ability did not reveal a significant effect of the reflective dialogues in either experiment. We had hoped that the capstone dialogues that treatment students completed before selected problems, at

the end of each unit, would support problem-solving ability. It is possible that the intervention consisted of too few capstone dialogues (only five) to observe any effect.

What seems to be the critical feature of post-practice reflection in supporting conceptual understanding is the explicit instruction that it provides in domain knowledge. PPR may help students to fill in knowledge gaps, resolve misconceptions, and abstract from the case at hand so that they are better prepared to engage in constructive activity (e.g., self-explanation) in future problems. Preliminary research comparing Andes (which encourages implicit learning of problem-solving strategies) with Pyrenees, a system that teaches problem-solving strategies explicitly, also suggests an advantage for explicit instruction [10]. Further research is needed to identify the mechanisms that drive learning from reflective dialogue, and to increase its potential to enhance problem-solving ability in addition to conceptual knowledge.

Acknowledgements. This research was supported by grants from the Office of Naval Research (grant number N00014-97-1-0848) and the Pittsburgh Science of Learning Center, which is funded by the National Science Foundation award number SBE-0354420. The data and views expressed are not necessarily endorsed by these agencies. We appreciate the collaboration of Professors Robert Shelby, Donald Treacy, and Mary Wintersgill throughout the development and testing of reflective dialogues. Kurt VanLehn offered many valuable suggestions. Pamela Jordan and Anders Weinstein provided crucial software support.

References

[1] Katz, S., & Allbritton, D., & Connelly, J. (2003). Going beyond the problem given: How human tutors use post-solution discussions to support transfer. *International Journal of Artificial Intelligence and Education*, *13* (1), 79-116.

[2] Lee, A. Y., & Hutchison, L. (1998). Improving learning from examples through reflection. *Journal of Experimental Psychology: Applied*, *4* (3), 187-210.

[3] VanLehn, K., Lynch, C., Schulze, K., Shapiro, J.A., Shelby, R., Taylor, L., Treacy, D., Weinstein, A., & Wintersgill, M. (2005). The Andes physics tutoring system: Lessons learned. *International Journal of Artificial Intelligence and Education*, *15* (3).

[4] VanLehn, K., Lynch, C., Schulze, K. Shapiro, J. A., Shelby, R., Taylor, L., Treacy, D., Weinstein, A., & Wintersgill, M. (2005). The Andes physics tutoring system: Five years of evaluations. In G. McCalla, C. K. Looi, B. Bredeweg & J. Breuker (Eds.), *Artificial Intelligence in Education* (pp. 678-685). Amsterdam, Netherlands: IOS Press.

[5] Aleven, V., & Koedinger, K. R. (2002). An Effective Meta-cognitive Strategy: Learning by Doing and Explaining with a Computer-Based Cognitive Tutor. *Cognitive Science*, *26* (2), 147-179.

[6] Halloun, I.A., & Hestenes, D. (1985). The initial knowledge state of college students. *The American Journal of Physics*, *53*, 1043-1055.

[7] Chi, M. T. H., Siler, S., Jeong, H., Yamauchi, T., & Hausmann, R. G. (2001). Learning from human tutoring. *Cognitive Science*, *25*, 471-533.

[8] Rosé, C. P., Bhembe, D., Siler, S., Srivastava, R. & VanLehn, K. (2003). Exploring the effectiveness of knowledge construction dialogues. In U. Hoppe, F. Verdejo, & J. Kay (Eds.), *Proceedings of the 11th International Conference on Artificial Intelligence in Education, AI-ED 2003* (pp. 497-499). Amsterdam: IOS Press.

[9] Dufresne, R.J., Gerace, P., Hardiman, T., & Mestre. (1992). Constraining novices to perform expertlike analyses: Effects on schema acquisition. The Journal of the Learning Sciences, 2 (3), 307-31.

[10] VanLehn, K., Bhembe, D., Chi, M., Lynch, C., Schulze, K., Shelby, R., Taylor, L., Treacy, D., Weinstein, A., & Wintersgill, M. (2004). Implicit versus explicit learning of strategies in a non-procedural cognitive skill. In J. C. Lester, R. M. Vicari, & F. Paraguacu, (Eds.), *Intelligent Tutoring Systems: 7th International Conference* (pp. 521-530). Berlin: Springer-Verlag Berlin & Heidelberg GmbH & Co.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Can a Polite Intelligent Tutoring System Lead to Improved Learning Outside of the Lab?

Bruce M. MCLAREN, Sung-Joo LIM, David YARON, and Ken KOEDINGER
Carnegie Mellon University
bmclaren@cs.cmu.edu, sungjol@andrew.cmu.edu, yaron@cmu.edu, koedinger@cs.cmu.edu

Abstract. In this work we are investigating the learning benefits of e-Learning principles (a) within the context of a web-based intelligent tutor and (b) in the "wild," that is, in real classroom (or homework) usage, outside of a controlled laboratory. In the study described in this paper, we focus on the benefits of politeness, as originally formulated by Brown and Levinson and more recently studied by Mayer and colleagues. We test the learning benefits of a stoichiometry tutor that provides polite problem statements, hints, and error messages as compared to one that provides more direct feedback. Although we find a small, but not significant, trend toward the polite tutor leading to better learning gains, our findings do not replicate that of Wang *et al*., who found significant learning gains through polite tutor feedback. While we hypothesize that an e-Learning principle such as politeness may not be robust enough to survive the transition from the lab to the "wild," we will continue to experiment with the polite stoichiometry tutor.

1. Introduction

The plethora of e-Learning materials available on the web today raises an important question: Does this new technology, delivered to students using new ways of presenting text, graphics, audio, and movies, lead to improved learning? It is important not only to provide easy access to learning technology but also to investigate scientifically whether that technology makes a difference to learning. Mayer and other educational technology researchers have proposed and investigated a variety of e-Learning principles, such as personalization (using conversational language, including first and second-person pronouns, in problem statements and feedback), worked examples (replacing some practice problems with worked examples), and contiguity (placing text near the graphics or pictures it describes), and have run a variety of (predominantly) lab-based experiments to test their efficacy [1]. The evidence-based approach taken by these researchers is important learning science research.

We are also interested in a systematic investigation of e-Learning principles. However, we wish to explore two aspects of e-Learning principles that have been left largely unexplored by previous researchers. First, our goal is to test the principles in the context of a web-based intelligent tutoring system (ITS), rather than in a standard e-Learning environment, which typically provides less feedback for and support of learners. Our main interest is in understanding whether and how the principles can supplement and extend the learning benefits of a tutoring system that runs on the web. Second, as a project within the Pittsburgh Science of Learning Center (PSLC) (www.learnlab.org), we wish to explore one of the key concepts underpinning the PSLC's theoretical framework and approach: the testing of learning interventions in the "wild" (i.e., in a live classroom or homework setting) rather than in a tightly controlled lab setting. The PSLC espouses an approach in which *in vivo* studies are the primary mechanism for experimentation; similar to the way studies are done in medical research

hospitals (http://www.learnlab.org/clusters/PSLC_Theory_Frame_June_15_2006.pdf).

We have initially focused our attention on two of the e-Learning principles mentioned above, in particular, personalization and worked examples. In two *in vivo* studies, in which we applied these principles in the context of an intelligent, web-based tutor, we were not able to replicate the learning benefits found by other researchers in more controlled, lab-based settings. Using a 2 x 2 factorial design, we found that personalization and worked examples had no significant effects on learning. The first study was conducted with 69 university students [2]. The second study, the results of which have not been published, was conducted with 76 students at two suburban U.S. high schools. In both of these studies there was a significant learning gain between the pre- and posttest across all conditions, in line with well-established findings from previous studies of the benefits of intelligent tutoring (e.g., [3, 4]).

So why didn't we achieve learning gains when adding these two e-Learning principles to a tutoring system? One possibility is that the tutoring may simply have had much more effect on learning than either the worked examples or personalization. The students learned at a significant level in both studies but much of that learning may have been induced by the support of the tutor. It is also possible that the principles did not survive the transition from the lab to *in vivo* experimentation. Most of the worked example experiments and all of the personalization experiments cited by Clark and Mayer [1] were conducted in tightly controlled lab environments of relatively short duration (< 1 hour), while our experiments were conducted in messier, real-world settings (i.e., the classroom or at home over the Internet) in which students used the tutor for 3 to 5 hours. It may also be that the principles only work for certain domains or for certain populations. For instance, some research has demonstrated that novices tend to benefit more from worked examples than more advanced students [5].

Another possible explanation, one that focuses on the personalization principle and is the main topic of the current paper, is that the personalization principle may need refinement and/or extension. In particular, perhaps our conceptualization and implementation of "personalization" was not as socially engaging and motivating as we had hoped. Mayer suggested this to us, after reviewing some of the stoichiometry tutor materials (personal communication) and based upon recent research he and colleagues have done in the area of *politeness* [6, 7, 8]. While our stoichiometry tutor is faithful to the personalization principle [1], it may be that the conversational style proposed by the principle is simply not enough to engage learners. Thus, we decided to "upgrade" the principle in its application to the stoichiometry tutor by including politeness. We did this by modifying all of the problem statements, hints, and error messages to create a new "polite" version of the stoichiometry tutor. We then executed a new *in vivo* experiment to test whether the new, polite version of the tutor led to greater learning gains than a more direct version.

In this paper, we briefly review the research on personalization and politeness in e-Learning, describe how we've created a polite version of the stoichiometry tutor, present and discuss the experiment we ran that tested whether the polite tutor improved learning, and conclude with hypotheses about our finding and proposed next steps.

2. Research on Personalization and Politeness in e-Learning

The personalization principle proposes that informal speech or text is more supportive of learning than formal speech or text in an e-Learning environment. E-Learning research in support of the personalization principle has been conducted primarily by Mayer and colleagues, with a total of 10 out of 10 studies demonstrating

deeper learning with personalized versions of e-Learning material, with a strong median effect size of 1.3 [9]. All of these studies focused on the learning of scientific concepts in e-Learning simulation environments and compared a group that received instruction in conversational style (i.e., the personalized group) with a group that received instruction in formal style (i.e., the nonpersonalized group). Note, however, that all of these studies were tightly controlled lab experiments with interventions of very short duration, some as short as 60 seconds, and none were conducted in conjunction with an intelligent tutoring system.

Based on the work of Brown and Levinson [10], Mayer and colleagues have more recently performed a series of studies to investigate whether "politeness" in educational software, in the form of positive and negative face saving feedback, can better support learners. They have implemented positive and negative face feedback in the context of the Virtual Factory Teaching System (VFTS), a factory modeling and simulation tutor. Positive face refers to people's desire to be accepted, respected, and valued by a partner in conversation, while negative face refers to the desire of people not to be controlled or impeded in conversation. In a *polite* version of VFTS they have developed, constructions such as, "You could press the ENTER key" and "Let's click the ENTER button" were used. Such statements are arguably good for positive face, as they are likely to be perceived as cooperative and suggestive of a common goal, as well as for negative face, as they are also likely to be perceived as respectful of the student's right to make his or her own decisions. In the *direct* version of VFTS, the tutor used more imperative, direct feedback such as, "Press the ENTER key" and "The system is asking you to click the ENTER button." These statements are arguably not supportive of positive face, as they do not suggest cooperation, or of negative face, as they are likely to be perceived as limiting the student's freedom.

In a preliminary study run by Mayer *et al.* [7] students were asked to evaluate the feedback of the tutor. The results indicated that learners are sensitive to politeness in tutorial feedback, and that learners with less computer experience react to the level of politeness in language more than experienced computer users. Wang *et al.* [6] ran students through a Wizard-Of-Oz study, with some students using the polite tutor and some using the direct tutor, that showed students liked working with the polite tutor more than the direct tutor and did slightly, but not significantly, better in learning outcome when using the polite tutor. Finally, in the study run by Wang *et al.* [8] in which 37 students were randomly assigned either to a polite tutor group or to a direct tutor group. The students who used the polite tutor scored significantly higher on a posttest. In sum, these studies suggest that it is not just first and second person conversational feedback, such as that proposed by the personalization principle and employed by the stoichiometry tutor, that makes a difference in motivating students and promoting better learning, but instead the level of politeness in that feedback. The Wang *et al.* study also showed that this effect could be achieved in the context of a tutor, something we are also interested in.

Not all research supports the idea that politeness will benefit learning and tutoring. For instance, Person *et al.* studied human tutoring dialogues and suggest that politeness could, under some circumstances and in different domains, inhibit effective tutoring [11]. They also relied on Brown and Levinson as a framework of investigation and found that different steps in the tutoring process appear to be more or less likely to benefit from politeness. However, these findings were not subjected to empirical study.

3. The Stoichiometry Tutor

The Stoichiometry Tutor, developed using the Cognitive Tutor Authoring Tools [12], and an example of a typical stoichiometry problem (a "polite" version) are shown in Figure 1. Solving a stoichiometry problem involves understanding basic chemistry concepts, such as the mole and unit conversions, and applying those concepts in solving simple algebraic equations. To solve problems with the tutor the student must fill in the terms of an equation, cancel numerators and denominators appropriately, provide reasons for each term of the equation (e.g., "Unit Conversion'), and calculate and fill in a final result. The tutor can provide student requested-hints, as is shown in Figure 1 (the hint refers to the highlighted cell in the figure), and also provides context-specific error messages when the student makes a mistake during problem solving [2].

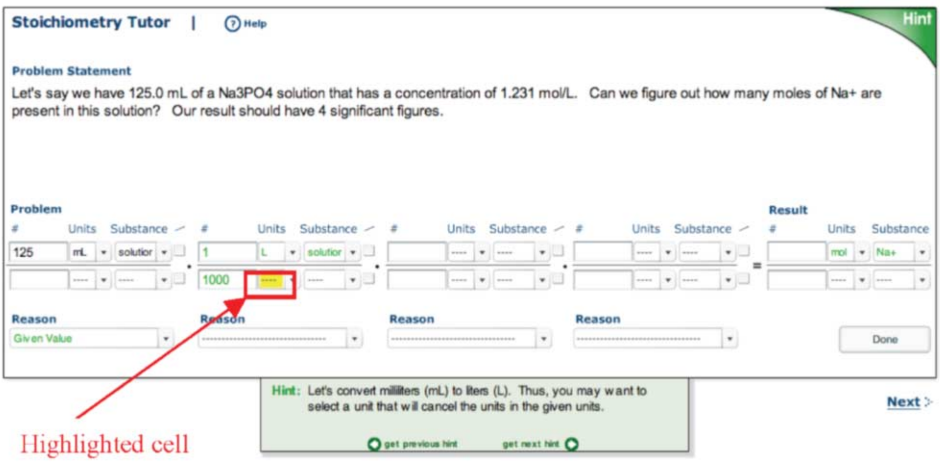


Figure 1: The Stoichiometry Intelligent Tutor

4. Changing the Stoichiometry Tutor From Personalization to Politeness

Given the learning outcomes achieved by Mayer and colleagues we decided to investigate whether altering our stoichiometry tutor's language feedback, making it more polite and improving its positive and negative face, would lead to better learning results. To implement and test this we created a "polite" version of the stoichiometry tutor by altering all of the problem statements, hints, and error feedback of the personalized version of the stoichiometry tutor to make them more polite, following the approach of Mayer *et al*. [7]. In addition, we added "success messages" to some correct responses made by students (in the personalized and direct versions of the tutor, there were no success messages), another positive polite strategy suggested in Wang *et al.* [6]. Examples of the changes we made to create a polite stoichiometry tutor, as well as how these changes compare to the direct and personal versions, are shown in Table 1.

Notice that all of the messages from the polite stoichiometry tutor are intended to be good for both positive and negative face, in comparison to the direct version of the stoichiometry tutor. For instance, the polite problem statement "Can we calculate the number of grams of iron (Fe) that are present in a gram of hematite (Fe2O3)?" provides positive politeness by suggesting cooperation and a common goal between the student and the tutor (use of "we") and negative politeness through giving the student freedom of choice (use of "Can" and phrasing the problem as a question; using "should" in the second sentence of the problem statement). In contrast, the analogous problem

statement in the direct stoichiometry tutor ("How many grams of iron (Fe) are present in a gram of hematite (Fe2O3)?") is lacking in both positive politeness (i.e., no sense of collaboration suggested) and negative politeness (i.e., the student's freedom of action is not acknowledged, as the wording of the statement assumes the student *will* do the problem). For similar reasons, all of the other problem statements, hints, and error feedback examples taken from the polite stoichiometry tutor in Table 1 have arguably more positive and negative politeness than the corresponding statements in the direct stoichiometry tutor. In addition, the polite version of the tutor is the only one to provide positive politeness in the form of success messages.

Table 1: Examples of Language Diffs. Between the Polite, Direct, and Personal Versions of the Stoich Tutor

	Polite Stoichiometry Tutor	**Direct Stoichiometry Tutor (“Impersonal” version, see [2])**	**Personal Stoichiometry Tutor (“Personal” version, see [2])**
Problem Stmts.	Can we calculate the number of grams of iron (Fe) that are present in a gram of hematite (Fe2O3)? Our result should have 5 significant figures.	How many grams of iron (Fe) are present in a gram of hematite (Fe2O3)? The result should have 5 significant figures.	Can you calculate and tell me how many grams of iron are present in a gram of hematite? Your result should have 5 significant figures.
Hints	Let's calculate the result now.	The goal here is to calculate the result.	You need to calculate the result.
	Do you want to put 1 mole in the numerator?	Put 1 mole in the numerator.	You need to put 1 mole in the numerator.
Error Msgs.	You could work on a composition stoichiometric relationship in this term. Are grams part of this relationship?	This problem involves a composition stoichiometric relationship in this term. Grams are not part of this relationship.	You need to work on a composition stoichiometric relationship in this term. Grams are not part of this relationship.
	Won't we need these units in the solution? Let's not cancel them, Ok?	No, these units are part of the solution and should not be cancelled.	Won't you need these units in the solution? If so you shouldn't cancel them.
Success Msgs.	Super job, keep it up.	*None*	*None*

5. Method

We conducted a study using the 2 x 2 factorial design discussed previously and shown in Table 2. One independent variable was politeness, with one level *polite* problem statements, feedback, and hints (i.e., use of the polite stoichiometry tutor) and the other level *direct* instruction, feedback, and hints (i.e., use of the direct stoichiometry tutor). The other independent variable was worked examples, with one level being *tutored only* and the other level *tutored and worked examples*. For the former level, subjects only solve problems with the tutor; no worked examples are presented. In the latter, subjects alternate between observation and (prompted) self-explanation of a worked example and solving of a problem with the aid of the tutor (either the polite or direct tutor, dependent on the condition). Note that although our main interest in this study was the effect of politeness, we decided to keep the design of the McLaren *et al.* experiment [2], since we were curious whether the null result with respect to worked examples in the previous experiment would hold again.

The study was conducted with 33 high school students at a suburban high school in Pittsburgh, PA. The materials were presented as an optional extra credit assignment in a college prep chemistry class, and the students could either do the work during a school

study hall or at home. Subjects were randomly assigned to one of the four conditions in Table 2.

Table 2: The 2 x 2 Factorial Design

	Polite Instruction	**Direct Instruction**
Tutored Only	Polite / Tutored (Condition 1)	Direct / Tutored (Condition 2)
Tutored and Worked Examples	Polite / Worked (Condition 3)	Direct / Worked (Condition 4)

All subjects were first given an online *pre-questionnaire* with multiple-choice questions designed to measure confidence (e.g., "I have a good knowledge of stoichiometry" Not at all, … Very Much) and computer experience (e.g., "How many hours a week do you normally use a computer?" < 1 hr, 1-5 hours, … > 20 hours). All subjects were then given an online pretest of 5 stoichiometry problems. The subjects took the pretest (and, later, the posttest) using the web-based interface of Figure 1, with feedback and hints disabled. The subjects then worked on 10 "study problems," presented according to the different experimental conditions of Table 2. All of the worked examples in the study were solved using the tutor interface, captured as a video file, and narrated by the first author (McLaren). During the solving of the 10 study problems, the subjects were also periodically presented with various instructional videos. After completing the 10 study problems, all subjects were given an online *post-questionnaire* with multiple-choice questions designed to assess their feelings about their experience with the tutor (e.g., "The tutor was friendly to me" Not at all … Almost Always). Finally, the subjects were asked to take a posttest of 5 problems, with problems isomorphic to the pretest.

All individual steps taken by the students in the stoichiometry interface of the pretest and posttest were logged and automatically marked as correct or incorrect. A score between 0 and 1.0 was calculated for each student's pretest and posttest by normalizing the number of correct steps divided by the total number of possibly correct steps.

6. Results

We did a two-way ANOVA (2 x 2) on the difference scores (posttest – pretest) of all subjects. There were no significant main effects of the politeness variable ($F(1, 29) = .66$, $p = .42$) or the worked examples variable ($F(1, 29) = .26$, $p = .61$). To test the overall effect of time (pretest to posttest), a repeated measure ANOVA (2 x 2 x 2) was conducted. Here there was a significant effect ($F(1, 29) = 48.04$, $p < .001$). In other words, students overall – regardless of condition – improved significantly from pretest to posttest. A summary of the results, in the form of a means of adjusted posttest, is shown in Figure 2. These results were very similar to the results of

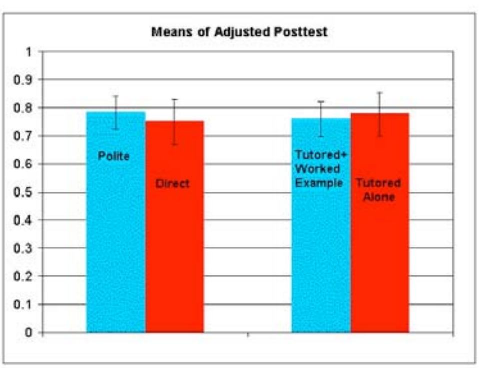


Figure 2: Means of Adjusted Posttest N=33.

McLaren et al. [2], except that in the current study the polite condition did somewhat better than the direct condition, as illustrated in Figure 2, whereas in the previous study the impersonal condition did slightly, but not significantly, better than the personal condition. While the difference is not significant, it nevertheless can be considered a small trend. We did a one-way ANOVA on posttest covariate with pretest that showed a small, but clearly larger, slope from the pretest to posttest for the polite condition.

Since politeness did not seem to support learning, contrary to the findings of Wang *et al*. [8], we did some analysis of the pre- and post-questionnaire responses, similar to Mayer *et al.* [7], to see if students at least noticed and reacted to the positive and negative politeness of the polite stoichiometry tutor's feedback. Students provided answers to the multiple-choice questions shown in the box below, selecting between 5 responses ("Not likely," "Not much," "Somewhat," "Quite a bit," and "Almost Always").

We scaled the responses from 1 to 5 and performed an independent-samples t-test between the polite and direct conditions. Only highlighted question 'g' led to a statistically significant difference between the polite and direct conditions (p=.000), with the polite condition having a higher rating than the direct condition. In other words, the polite stoichiometry tutor's feedback only appeared to be received as especially polite and helpful in how it praised students for doing something right.

a. I liked working with the stoichiometry tutor
b. The tutor helped me identify my mistakes
c. The tutor worked with me toward a common goal
d. The tutor was friendly to me
e. The tutor let me make my own choices
f. The tutor made me follow instructions
g. The tutor praised me when I did something right
h. The tutor was critical of me
i. My relationship with the tutor was improving over time

While we did not find a strong overall effect of subjects finding the polite tutor more polite than the direct tutor, we did find certain groups of students who were sensitive to the more polite statements. For instance, within a "low-confidence group" (i.e., N = 9; subjects who had a score <=5 when summing 3 confidence questions, highest possible score = 15), we found a stronger sensitivity to the polite tutor's statements. The low-confidence students gave higher ratings, at a marginally significant level, to the positive politeness questions e ("The tutor let me make my own choices," p=.09) and c ("The tutor worked with me toward a common goal," p=.119) and lower ratings, again at a marginally significant level, to the negative politeness question f ("The tutor made me follow instructions," p=.112). Within a "low-skill group" (i.e., N = 8; subjects who scored less than 0.5 (50%) on the pretest), we also found positive sensitivity to the polite tutor's statements. The low-skill students gave higher ratings, at a significant level, to positive politeness question i ("My relationship with the tutor was improving over time," p=.041) and higher ratings at a significance level that, while not marginally significant, were quite a bit lower than all others to the positive politeness questions a ("I liked working with the stoichiometry tutor," p=.165) and b ("The tutor helped me identify my mistakes," p=.165).

7. Discussion and Conclusions

Although the polite stoichiometry tutor showed a small trend toward better learning gains than a more direct version of the tutor, our findings, at least to this point, do not support the findings of Wang *et al.* [8] that politeness makes a difference in learning.

There are a variety of possible reasons for this, but our primary hypothesis is that politeness, while making a difference in controlled lab studies of short duration is less

likely to have an effect in a real world study such as ours. One might argue that such a study is subject to methodological problems, such as excessive time on task, etc., but our objective, as part of a learning science center with a particular mission, is to test interventions in just these kinds of circumstances. After all, if educational interventions cannot endure the transition from the lab to the real world, than their usefulness is questionable. As demonstrated in our two studies thus far, the use of an intelligent tutor *did* make a significant difference to learning in a classroom / homework situation; it is the additional e-Learning principles that did not appear to add to learning gains.

Another possibility is that our polite tutor didn't exhibit enough positive and negative politeness to have a real impact on students. There is some evidence of that in our analysis of the post-questionnaire. While there is some evidence that students who used the polite tutor really found it more polite, especially within certain groups such as low-confidence and low-skill students, the overall impression of politeness it left on students does not appear to rise to the level reported in [7]. Our earlier hypothesis that both of the e-Learning principles were "swamped" by the effect of tutoring is potentially refuted by Wang *et al* [8]. In their study, with an N very similar to ours, the effect of tutoring did not overwhelm the effect of politeness. That is, in their study, politeness made a difference even in the midst of tutoring.

In conclusion, however, both our N of 33 and theirs of 37 are relatively small; a power analysis we performed indicated the need for an N = ~320 to reach a power of .8. In an attempt to get a larger effect size and more fully investigate the polite version of the stoichiometry tutors, we are planning a second phase of this study at two suburban New Jersey high schools in early 2007.

Acknowledgements. The PSLC, NSF Grant # 0354420, has provided support for this research.

References

[1] Clark, R. C. & Mayer, R. E. (2003). *e-Learning and the Science of Instruction*. Jossey-Bass/Pfeiffer.

[2] McLaren, B. M., Lim, S., Gagnon, F., Yaron, D., & Koedinger, K. R. (2006). Studying the Effects of Personalized Language and Worked Examples in the Context of a Web-Based Intelligent Tutor; In the *Proceedings of the 8th International Conference on Intelligent Tutoring Systems*, 318-328.

[3] Koedinger, K. R., Anderson, J. R., Hadley, W. H., & Mark, M. A. (1997). Intelligent tutoring goes to school in the big city. *International Journal of Artificial Intelligence in Education*, 8, 30-43.

[4] VanLehn, K., Lynch, C., Schulze, K., Shapiro, J., & Shelby. (2005). The Andes Physics Tutoring System: Five Years of Evaluations. In the *Proceedings of AIED-05.* 678-685.

[5] Kalyuga, S., Chandler, P., Tuovinen, J., & Sweller, J. (2001). When problem solving is superior to studying worked examples. *Journal of Ed. Psych., 93*, 579-588.

[6] Wang, N., Johnson, W. L., Rizzo, P., Shaw, E., & Mayer, R. E. (2005). Experimental Evaluation of Polite Interaction Tactics for Pedagogical Agents. In the *Proceedings of the 2005 International Conference on Intelligent User Interfaces,* 12-19. ACM Press: New York.

[7] Mayer, R. E., Johnson, W. L., Shaw, E. & Sandhu, S. (2006). Constructing Computer-Based Tutors that are Socially Sensitive: Politeness in Educational Software. *Int. J. Human-Computer Studies,* 64, 36-42.

[8] Wang, N., Johnson, W. L., Mayer, R. E., Rizzo, P., Shaw, E., & Collins, H. (in press). The Politeness Effect: Pedagogical Agents and Learning Outcomes. To be published in the *International Journal of Human-Computer Studies*.

[9] Mayer, R. (2005). Principles of Multimedia Learning Based on Social Cues: Personalization, Voice, and Image Principles. *The Cambridge Handbook of Multimedia Learning*, R. E. Mayer (Ed). Chapter 14, 201-212.

[10] Brown, P. & Levinson, S. C. (1987). *Politeness: Some Universals in Language Use*. Cambridge University Press, New York.

[11] Person, N. K., Kreuz, R. J., Zwaan, R. A., & Graesser, A. C. (1995). Pragmatics and Pedagogy: Conversational Rules and Politeness Strategies May Inhibit Effective Tutoring. *Cognition and Instruction* 13, 161-188.

[12] Koedinger, K. R., Aleven, V., Heffernan, N., McLaren, B. M., & Hockenberry, M. (2004) Opening the Door to Non-Programmers: Authoring Intelligent Tutor Behavior by Demonstration; *Proceedings of ITS-2004*, 162-174.

Authoring Tools and Ontologies

This page intentionally left blank

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Lowering the Bar for Creating Model-Tracing Intelligent Tutoring Systems

Stephen BLESSING[a,1], Stephen GILBERT[b,c], Steven OURADA[b], Steven RITTER[d]
[a] *Department of Psychology, University of Tampa, Tampa, FL*
[b] *Clearsighted, Inc.;* [c] *ISU Center for Technology;* [d] *Carnegie Learning, Inc.*

Abstract. The main goal of the work presented here is to allow for the broader dissemination of intelligent tutoring technology. To accomplish this goal, we have two clear objectives. First, we want to allow different types of people to author model-tracing intelligent tutoring systems (ITSs) than can now do so. Second, we want to enable an author to create a tutor for software that was not initially designed with an ITS in mind. Accomplishing these two objectives should increase the number of such ITSs that are produced. We have created the first iteration of an authoring system that addresses both objectives. Non-cognitive scientists and non-programmers have used the system to create a tutor, and the system can interface with third-party software that was not originally designed with the ITS.

1. Introduction

Intelligent Tutoring Systems (ITSs) have been shown effective in a wide variety of domains and situations. Within the different types of ITSs, model-tracing tutors have been particularly effective [1, 4, 13]. These tutors contain a cognitive model of the domain that the tutor uses to check most, if not all, student responses. This type of intense interaction and feedback has led to impressive student learning gains. Results show that students can master the material in a third less time [3]. Field trials comprising hundreds of students have shown significant improvement on standardized tests when students practiced with model-tracing tutors [6].

Despite these successes, ITSs, including model-tracing tutors, have not been widely adopted in educational or other settings, such as corporate training. Perhaps the most successful deployment of model-tracing tutors is Carnegie Learning's Cognitive Tutors for math, which is in use by over 1000 school districts by hundreds of thousands of students. After this notable success, however, most educational and training software is not of the ITS variety, let alone a model-tracing ITS, in spite of the impressive learning advantages. There are two clear and related reasons for this lack of adoption: the expertise needed to create such a system and the costs involved.

Both of these issues, expertise and cost, are connected to the fact that to create a complete model-tracing ITS requires many different software pieces: an interface, a curriculum, a learner-management system, a teacher report package, and the piece that provides the intelligence, a cognitive model. To create these pieces, many kinds of expertise are needed, most notably programming (both GUI and non-GUI), pedagogy,

[1] Corresponding Author: Department of Psychology, University of Tampa, 401 W. Kennedy Blvd. Box Q, Tampa, FL 33606 USA; e-mail: sblessing@ut.edu

and cognitive science. This generally requires a team of highly trained people to work together, thereby resulting in a high cost. Past estimates have found that it takes 100 hours or more to create 1 hour of instruction [7]. Taken together, these factors of expertise and cost obviously are a large part of why model-tracing ITSs are not in wider use.

The work presented here attempts to decrease these factors for creating model-tracing ITSs by reducing the expertise needed to create such a tutor, which will result in reducing the cost as well. Our solution for this "lowering the bar" of ITS creation consists of two parts: simplifying cognitive model creation with an easier-to-use authoring system and linking existing software interfaces to a cognitive model.

Simplifying Cognitive Model Creation. The aspect of a model-tracing tutor that provides its unique sense of student interaction, the cognitive model, has required arguably the most expertise to produce. In the past to create these cognitive models, which often take the form of a production system, one needed a high level of competence in both cognitive science and computer programming. This relegated their creation to mostly Ph.D. cognitive scientists who have specialized in such endeavors. Recent research by various investigators has attempted to create authoring systems that make creating ITSs, particularly the aspect that makes them intelligent, easier (see [8] for a review).

The plan we have for meeting this objective is to streamline the process for creating a cognitive model, and to eliminate, or at least reduce greatly, the programming aspect. An author who wants to create a cognitive model should not be required to be a programmer. Indeed, our ultimate aim is to allow a motivated master teacher to be able to create the intelligence behind an ITS, or at the very least modify in a meaningful way an already produced cognitive model.

Linking Existing Software Interfaces. To create an ITS, one needs not only to produce the cognitive model, but also the student interface as well. Creating this is akin to creating any other piece of software, which adds to the expertise and cost of the overall system. If the ITS team could link a cognitive model to an off-the-shelf piece of software with little or no modification, then this would result in a huge cost and time savings.

The plan we have here is to be able to integrate an already produced GUI with the cognitive model created in the manner as described above. To do this we will concentrate on using already developed protocols for communication with the GUI, but we will also consider other solutions as well.

What follows are two main sections that explain more deeply our two-part solution to lowering the bar in creating model-tracing ITSs. The first part discusses our Cognitive Model Software Development Kit (SDK), followed by an original description of our system for leveraging existing interfaces, a feature found in few other systems. Within both sections are discussions of our observations and new empirical findings of actually using the system in our work.

2. The Cognitive Model SDK

In order to lower the bar in creating the cognitive model component of an ITS, we have concentrated on creating and using representations that are easier to understand than the working and procedural memory representations used by past cognitive model development environments. In doing so, we hope to enable non-cognitive scientists and

non-programmers to create cognitive models. This goal is similar to another project [5] in their effort to lower the bar for creating cognitive models. However, we are approaching the task from a different direction. Their efforts emphasize programming-by-demonstration and other techniques beneficial to non-programmers, whereas our system uses more complex representations. There is an obvious trade-off between ease-of-use and the expressiveness of the system, but by pursuing both ends we can find the common ground of these two approaches. Our interests are driven by the need to support already exiting tutors in use by hundreds of thousands of students, which places additional needs of maintainability and robustness on our system.

Within the cognitive models for our model-tracing ITSs, there are two main components: an object model and a rule hierarchy. The object model contains the parts of the domain relevant to tutoring, and this object model serves as the basis of the rule hierarchy to provide the tutoring (e.g., hints and just-in-time messages) to the student. These two constructs are at the heart of our tutoring architecture. The particulars of the internal architecture used by us, referred to the as the Tutor Runtime Engine (TRE), has been described elsewhere [10]. A set of tools has been created that work on top of this architecture to provide a SDK for cognitive models. The next two sections provide a summary of how the object model and the rule hierarchy are created within the cognitive model SDK (see [2] for a fuller discussion).

For the object model and rule hierarchy editors, the key is finding representations that are understandable by non-programmers yet still provide enough power to create useful ITSs. We sought interfaces that have been successful in other software applications, such as tree views, hierarchies, and point-and-click interfaces. Other ITS SDKs have started using these and others to some degree (the VIVIDS system [9] was a particular inspiration to us).

2.1 Object Model Editor

An important aspect of a model-tracing ITS cognitive model is the declarative structure that is used to refer to the important aspects of the domain and interface during tutoring. As opposed to our previous development environment for a domain's object model (a blank source document in Macintosh Common Lisp), this part of the SDK requires no programming knowledge, thereby lowering the bar considerably and results in a much more understandable, and maintainable, system. As an aside, both the object model and the rule hierarchy editors are capable of editing the current cognitive models in use by the commercial versions of Carnegie Learning's Cognitive Tutor, thereby demonstrating their robustness.

The object model editor does similar things as already existing tools (e.g., Protégé, an open-sourced ontology editor developed at Stanford University). The basic functionality is to display and edit objects consisting of attribute/value pairs. Our cognitive models, however, contain specific items with particular functionality that made it more attractive to create our own editor. Specifically, pre-defined object types exist that have their own properties and behaviors, and this has ramifications for how the rest of the system operates. For example there is a ***goalnode*** object type (representing a student's subgoal in the problem) that has a set of predefined properties, and attached to these goalnode types is a predicate tree inspector (the subject of the other main tool, representing the rule hierarchy).

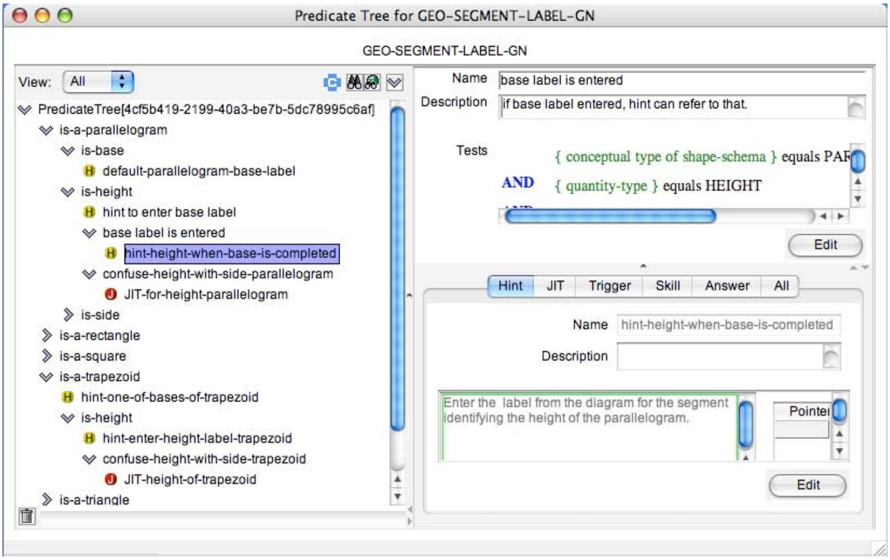


Figure 1. *Rule Hierarchy Editor.*

2.2 Rule Hierarchy Editor

The piece of a cognitive model that provides for the hints, just-in-time messages, and other important tutoring behavior is the set of rules that work with the object hierarchy. In our past system, these rules were realized by a production system, which required not only knowledge of cognitive science but also programming. Again, the challenge is to provide something that is powerful enough to meet our requirements for expressiveness, but is still usable by non-Ph.D. cognitive scientists. Our rule hierarchy editor (what we call a predicate tree) uses a hierarchical tree view, where each branch contains a predicate relating to the current object of interest (a goalnode, in our parlance), and most of the actions (the hints, for example), are contained at the leaf nodes. Figuring out which hint to give, then, amounts to sorting down through the predicate tree to see what matches the current set of conditions for the current object.

Figure 1 shows the current state of the tool (the Object Model Editor has a similar look, with objects and properties in a left-hand pane, and a right-hand pane that shows current values). The upper left pane of the design shows the predicate tree hierarchy for a particular goalnode. The upper right pane shows the full set of predicate tests for the selected node at the left, and the lower right pane shows the hints, just-in-time messages, and other tutoring actions attached at this node. Not shown is the rule editor that enables the creation and editing of nodes on the predicate tree.

2.3 Usability by Non-Cognitive Scientists and Non-Programmers

We have had some experience now using our SDK in various settings. What follows are three indications that the cognitive model SDK and the kinds of representations it uses are indeed usable by non-cognitive scientists and non-programmers.

Usability by non-cognitive scientists. At the very least, we want the SDK to be usable by non-cognitive scientists, so that other people may both create new cognitive

models and maintain existing ones. For example, the creation of a new cognitive model or the extensive modification of an existing one had to be done by a cognitive scientist. Furthermore, it would have been impossible for a curriculum designer to make just a simple wording change in one of the hints. Even that simple maintenance task had to be done by a cognitive scientist. Allowing other kinds of people to make these additions and modifications has obvious cost and time benefits.

In late 2005 and early 2006, two non-cognitive scientists at Carnegie Learning re-implemented parts of their geometry curriculum. They recreated around 25 hours of curriculum instruction, and it took them about 600 hours working with an early version of the SDK. While the 24 hours per hour instruction is quite good, and we are pleased with it, there are obvious provisos. While non-cognitive scientists, they were programmers that had been working and observing cognitive scientists. Also, they were recreating the cognitive model, not making a new one. Finally, that is just the time for creating the cognitive model and a few problems—the interface and some number of the problems had already been done. However, because it was an early version of the SDK, they were adding new features and fixing bugs to the system as they worked on the model itself. While this all makes the effort hard to interpret, we do take it as evidence that a commercial-quality cognitive model can be created via the SDK by non-cognitive scientists.

Usability by non-programmers. There are two pieces of evidence that one doesn't even need programming knowledge to use the SDK. The first was described elsewhere [2] and concerns an experiment investigating whether non-programmers could make sense of the representations used in the SDK. In this experiment, non-computer science undergraduates (nor were any cognitive science undergraduates) demonstrated a high degree of facility and competence at using the hierarchical views used by both the object model and rule hierarchy editors. This gave us confidence that the SDK was at least in the right direction regarding how it represented cognitive models.

We have since gone a step further and have actually had beginning cognitive modelers use the SDK to produce a cognitive model. Clearsighted, Inc. uses student interns as instructional designers and has developed a brief training curriculum to orient them to the SDK and the process of cognitive modeling. During Fall 2006 this curriculum was used successfully to enable two non-programmer graduate students from instructional design and one computer science undergraduate to complete a cognitive model for multi-column addition. They have since assisted in other cognitive modeling tasks. Although somewhat anecdotal, we again take this as encouragement that we have lowered the bar in creating cognitive models. Using the older system, this kind of result with non-programmers would simply not be possible.

3. Use of Existing Interfaces

Almost all Cognitive Tutor interfaces have been designed alongside the cognitive model (though see [12] for description of tutors that use Microsoft Excel and Geometer's Sketchpad). It is desirable to enable the construction of model-tracing ITSs around pre-existing software. This would eliminate or at least greatly diminish the time spent doing traditional interface programming. By allowing authors to create an ITS for off-the-shelf software, this will obviously lower the bar for creating such systems. One can imagine tutoring not only math or statistics using Microsoft Excel as the interface (or some other spreadsheet), but one could also tutor on Microsoft Excel itself (or any

other application requiring training). Such a scenario would be a boon to corporate training environments.

What is needed is a way for the Tutor Runtime Engine, the tool that uses the cognitive model created by the SDK, to communicate with third-party software (that is, software that the authors creating the ITS did not program). Even though the interfaces for the current Cognitive Tutors were developed essentially in tandem with their cognitive models, the code for the interfaces was separate from the tutoring code, with the two pieces communicating to each other through a messaging protocol (this is described in more depth in [10]). This separation allows for the kind of SDK described in the previous section, and also allows for the possibility of third-party applications to be the student interface. TutorLink is our solution for allowing the TRE to communicate with outside applications (see Figure 2).

3.1 TutorLink

In order to provide tutoring, the TRE engine needs to know what the student is doing. Either the interface needs to communicate what is happening or those actions must be inferred by some mechanism. There are three basic ways by which this might occur. First, the developer of the interface application uses tutorable widgets (that is, buttons, entry boxes, menus, etc.) that readily broadcast what they are doing in a way that is easily and straightforwardly understood by the TRE. This is what the current interfaces used within the Cognitive Tutors developed by Carnegie Learning do. Second, and least desirable of the three, occurs when the application is truly a black box, but based on low-level operating system events inferences are made as to what buttons, menus, entry boxes, and other widgets are being manipulated. You could do this with any application, but this solution is obviously very brittle, as any change to the application will probably break the tutoring interaction. The third choice is when the interface developer didn't use tutorable widgets, and the source code is not available to be recompiled with tutorable widgets. However, there may still be a way, that does not involve low-level OS events, to know what widgets the user is manipulating and map that to something with which the TRE can work. Different operating systems and architectures have mechanisms like this to different degrees. On the Macintosh platform, AppleEvents, if implemented correctly by the programmer, can provide excellent semantic-level events with which to use for tutoring [see [11] for a pre-TutorLink implementation of this mechanism). On Microsoft platforms, both Microsoft Foundation Classes and their .NET architecture allow for outside programs at least some insight and control as to what is happening in the interface. Java also allows for this to at least some degree. It is Microsoft's .NET architecture with which we have been currently experimenting. Depending on the exact nature of the representation, this third option is more or less brittle, and so is more advantageous than the second option.

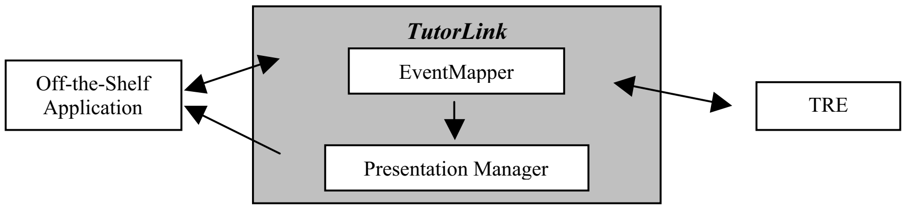


Figure 2. *TutorLink Architecture*

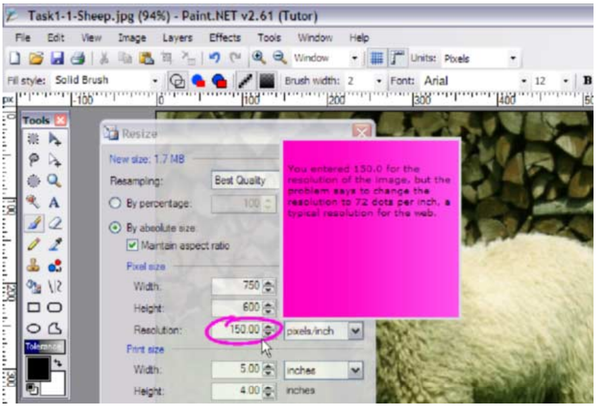


Figure 3. *Paint.Net as a tutored application.*

Regardless of how exactly TutorLink is monitoring the application that contains the interface, once the user interacts with the interface, that interaction needs to be noted by TutorLink and sent to the TRE for the appropriate tutoring action (the TRE is a java process; TutorLink could be anything). The part of TutorLink that receives the interaction message from the interface is called the EventMapper. As its name implies, this part takes the incoming message and maps the event into something the TRE can understand (that is, a goalnode within the cognitive model). In the case where the interface was built using tutorable widgets, this mapping is straightforward. In other cases, there needs to be a more structured mapping table that maps between interface widgets and goalnodes (e.g., both 'color wheel output' and 'list of red, green, blue textbox values' map to the 'answer for the choose-color goalnode). This mapping has to be supplied by the author. Such a mapping application using AppleEvents has been described [11]. In general terms, the author starts a listening application, then begins to interact with the widgets in the interface in order to identify their names, and then maps those to goalnodes. We developed such an application for the .NET architecture, and by extension, one could be developed for other architectures as well.

Once the event has been mapped for the tutor by the EventMapper, the message is transferred to the TRE. The tutor will decide on the appropriateness of the user action, and offer feedback (be it a hint, a just-in-time message, a change to the skill window, or something else) that will be communicated back to the interface through TutorLink via the EventMapper. A part of TutorLink referred to as the PresentationManager will decide how to present the tutor's feedback to the user. Again, depending on which architecture is being used, the feedback might be able to be presented within the interface, such that the user would not realize that a different program is really presenting the information (possible when using tutorable widgets, AppleEvents, and .NET), or the feedback might need to be presented within its own process.

3.2 Current Status

As mentioned above, we currently have a system using this architecture within Microsoft's .NET framework. Figure 3 shows a screenshot of our system in action (the tutor has indicated that the entered value of 150 for resolution is incorrect, and is providing a just-in-time message). The tutoring backend is the same backend that Carnegie Learning uses for its algebra tutors (that is, the TRE Engine). The frontend is an application called Paint.NET, image manipulation software akin to Adobe

Photoshop. We have developed a curriculum around Paint.NET that teaches several lessons with a model-tracing ITS. We are currently testing its effectiveness. Paint.NET is obviously a piece of software that was not designed with a model-tracing tutor in mind, so being able to provide such a rich tutoring experience within it has been the culmination of much previous work.

4. Discussion

We have a described a system that lowers the bar for creating model-tracing ITSs. The system achieves this goal by accomplishing two main objectives: 1) providing a feasible way in which non-programmers and non-cognitive scientists can create cognitive models; and 2) providing a mechanism by which third-party applications can be instrumented and used as the frontend of a tutoring system. Meeting both objectives lowers the bar necessary to create this effective class of tutor. Current effort is focused on ensuring that the workflow presented by the SDK assists both novice and expert users of the system, as well as enlarging the classes of systems with which TutorLink will work.

5. Notes

This material is based upon work supported by the National Science Foundation under Grant No. OII-0548754. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

6. References

[1] Anderson, J. R., Conrad, F. G., & Corbett, A. T. (1989). Skill acquisition and the LISP Tutor. *Cognitive Science*, *13*, 467-506.

[2] Blessing, S. B., Gilbert, S, & Ritter, S. (2006). Developing an authoring system for cognitive models within commercial-quality ITSs. In *Proceedings of the Nineteenth International FLAIRS Conference*.

[3] Corbett, A.T. (2001). Cognitive computer tutors: Solving the two-sigma problem. In the *Proceedings of the Eighth International Conference of User Modelling.*

[4] Koedinger, K. R., Anderson, J. R., Hadley, W. H., & Mark, M. A. (1997). Intelligent tutoring goes to school in the big city. *International Journal of Artificial Intelligence in Education, 8*, 30-43.

[5] Koedinger, K. R., Aleven, V., Heffernan, N., McLaren, B. M., & Hockenberry, M. (2004). Opening the Door to Non-Programmers: Authoring Intelligent Tutor Behavior by Demonstration. In the *Proceedings of the Seventh International Conference on Intelligent Tutoring Systems*.

[6] Morgan, P., & Ritter, S. (2002). An experimental study of the effects of Cognitive Tutor® Alegbra I on student knowledge and attitude. [http://www.carnegielearning.com/research/research_reports/morgan_ritter_2002.pdf].

[7] Murray, T. (1999). Authoring Intelligent Tutoring Systems: An analysis of the state of the art. *International Journal of AI in Education (1999), 10*, 98-129.

[8] Murray, T., Blessing, S., & Ainsworth, S. (2003). *Authoring tools for advanced technology educational software*. Kluwer Academic Publishers.

[9] Munro, A., Johnson, M.C., Pizzini, Q.A., Surmon, D.S., & Wogulis, J.L. (1996). A tool for building simulation-based learning environments. In *Simulation-Based Learning Technology Workshop Proceedings*, *ITS96*.

[10] Ritter, S., & Blessing, S. B., Wheeler, L. (2003). User modeling and problem-space representation in the tutor runtime engine. In P. Brusilovsky, A. T. Corbett , & F. de Rosis (Eds.), *User Modeling 2003* (pp. 333-336). Springer-Verlag.

[11] Ritter, S. & Blessing, S. B. (1998). Authoring tools for component-based learning environments. *Journal of the Learning Sciences*, *7*(*1*), 107–131.

[12] Ritter, S., & Koedinger, K. R. (1996). An architecture for plug-in tutor agents. *Journal of Artificial Intelligence in Education*, 7, 315-347.

[13] VanLehn, K., Lynch, C., Schulze, K., Shapiro, J.A., Shelby, R., Taylor, L., Treacy, D., Weinstein, A., & Wintersgill, M. (2005). The Andes physics tutoring system: Lessons learned. *International Journal of Artificial Intelligence and Education*, *15*(*3*).

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Constraint Authoring System: An Empirical Evaluation

Pramuditha SURAWEERA, Antonija MITROVIC and Brent MARTIN

Intelligent Computer Tutoring Group
Department of Computer Science, University of Canterbury
Private Bag 4800, Christchurch, New Zealand
{pramudi, tanja, brent}@cosc.canterbury.ac.nz

Abstract. Evaluation is an integral part of research that provides a true measure of effectiveness. This paper presents a study conducted to evaluate the effectiveness of CAS, a knowledge acquisition authoring system developed for generating the domain knowledge required for constraint-based tutoring systems with the assistance of a domain expert. The study involved a group of novice ITS authors composing domain models for adding two fractions. The results of the study showed that CAS was capable of generating highly accurate knowledge bases with considerably less effort than the effort required in manual composition.

Introduction

Composing the domain knowledge required for Intelligent Tutoring Systems (ITS) consumes the majority of the total development time [6]. The task requires a multi-faceted set of expertise, including knowledge engineering, AI programming and the domain itself. Our goal is to widen the knowledge acquisition bottleneck by empowering domain experts with little or no programming and knowledge engineering expertise to produce domain models necessary for ITSs.

Researchers have been exploring ways of automating the knowledge acquisition process since the inception of ITSs. Disciple, developed by Tecuci and co-workers[10], is an example of a learning agent shell for developing intelligent educational agents. A domain expert teaches the agent to perform domain-specific tasks by providing examples and explanations (similar to an expert teaching an apprentice), and Disciple uses machine learning techniques to infer the necessary domain knowledge. The Cognitive Tutor Authoring Tools (CTAT) [1] assist in the creation and delivery of model-tracing ITSs. The main goal of these tools is to reduce the amount of artificial intelligence (AI) programming expertise required. CTAT allows authors to create two types of tutors: 'Cognitive tutors' and 'Pseudo tutors'. 'Cognitive tutors' contain a cognitive model that simulates the student's thinking to monitor and provide pedagogical assistance during problem solving. In contrast, 'Pseudo tutors' do not contain a cognitive model: to develop a tutor of this kind, the author needs to record possible student actions and provide corresponding feedback messages.

We have developed an authoring system, named Constraint Authoring System (CAS) that generates a domain model with the assistance of a domain expert. The author is required to describe a domain in terms of an ontology and provide problems

and their solutions. CAS analyses the provided information using machine learning techniques to generate a domain model.

This paper outlines a study conducted with a group of novice authors to evaluate the effectiveness of CAS. They were given the task of composing a domain model for a fractions addition ITS using CAS. The results showed that CAS was capable of generating highly accurate constraint bases even with the assistance of novices. It also showed that CAS reduced the overall effort required to produce domain models.

The remainder of the paper is organised into three sections. The next section gives a brief introduction to CAS. The results and analysis of the experiment are given in section three. The final section presents conclusions and future work.

1. Constraint Authoring System

Constraint Acquisition System is an authoring system that generates the domain model required for constraint-based tutoring systems [5] with the assistance of a domain expert/ teacher. The goal of the system is to significantly reduce the time and effort required for composing constraint bases. We envisage that CAS would enable domain experts with minimal expertise in constraint-based modelling to produce new tutoring systems. The user is only required to model the domain in terms of an ontology and provide example problems and their solutions. Once the required domain-dependant information is provided, CAS generates constraints by analysing the provided information. Although the system is designed to support authors with minimal knowledge engineering expertise, it also offers utilities for experts in composing constraint bases. The system allows experts to modify constraints during the stage of validating the system-generated constraints. Users are also provided with editors for directly adding new constraints to the domain model.

A detailed discussion of CAS is beyond the scope of this paper and has been presented in [7]. Here we only give a short overview of its features. CAS is developed as an extension of WETAS [3], a web-based tutoring shell that facilitates building constraint-based tutors. WETAS provides all the domain-independent components for a text based ITS, including the user interface, pedagogical module and student modeller. It does not provide support for authoring domain models; however, authoring tools may be added to WETAS to provide this need. CAS was used in this manner.

Authoring knowledge using CAS is a semi-automated process with the assistance of a domain expert. The author carries out a number of tasks, including modelling the domain as an ontology, specifying the general structure of solutions and providing example problems and solutions. The constraint generators of CAS use this information to generate both syntax and semantic constraints. The syntax constraints are generated by analysing the ontology and translating each specified restriction into a constraint [9]. The semantic constraints generator uses machine learning techniques to generate constraints by analysing the problems and solutions provided by the domain expert [7]. It generates constraints by comparing and contrasting two alternative solutions to the same problem. Constraints are generated iteratively, and they are generalised or specialised during subsequent analysis of other solutions.

The interface of CAS consists of two main tabs: the “ontology view” and the “domain model editor”. The “ontology view” contains the ontology workspace and other tools necessary for composing the domain-dependant components necessary for each step of the authoring process. Figure 1 illustrates the ontology for the fraction

addition domain, and the relationships defined for the *Improper Fraction* concept. The “ontology view” also contains an interface for adding problems and their solutions that are used for constraint generation. The “domain model editor” tab consists of a set of textual editors for viewing and modifying constraints generated by CAS. Constraints in these editors can be directly loaded into WETAS for testing in an ITS environment. The “domain model editor” tab also contains an editor for modifying and composing problems and solutions in the Lisp representation expected by WETAS.

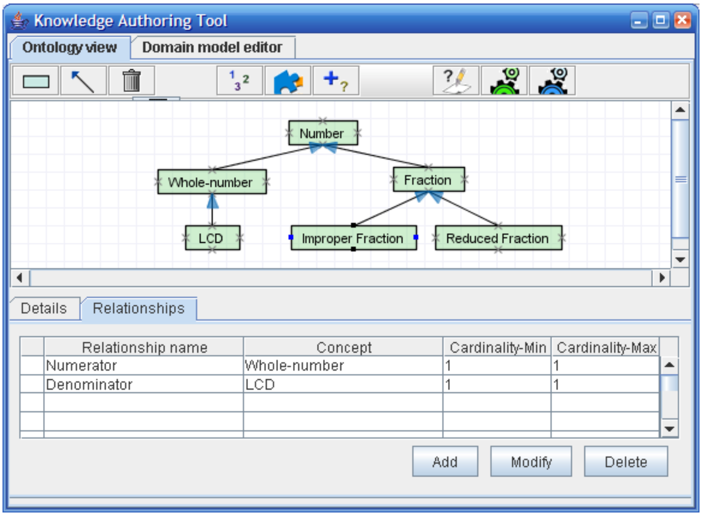


Figure 1: CAS Interface, illustrating the ontology workspace

2. Evaluation Study

In previous work we evaluated CAS by comparing its generated domain models to the domain models developed manually [7, 9]. The domain models were generated with the assistance of expert authors. However, the goal of CAS is to support novice authors to develop ITSs. For that reason, we performed a study with a group of 13 novice authors to evaluate the effectiveness of the system as a whole. The participants were students enrolled in a 2006 graduate course about ITSs at the University of Canterbury. They were assigned the task of producing a complete ITS in WETAS for the domain of adding fractions, using CAS to author the domain model. The goal of the evaluation study was to validate three hypotheses: CAS makes it possible for novices to produce complete constraint bases; the process of authoring constraints using CAS requires less effort than composing constraints manually; the constraint generation algorithm depends on the ontology (i.e. an incomplete ontology will result in an incomplete constraint set).

The participants were required to compose all the domain-dependant components necessary for CAS to generate constraints, including a domain ontology, problems and solutions. After composing the required components, CAS can be used to generate syntax and semantic constraints in a high-level language (pseudo-code). At the time of the study, CAS was not able to convert these into the WETAS constraint language, so the participants were also required to perform this step manually. They were also free to modify/delete constraints.

The participants had attended 13 lectures on ITSs, including five on constraint-based modelling before the study. They were introduced to CAS and WETAS, and were given a task description document that outlined the process of authoring a domain model using CAS and described the WETAS constraint language. The participants were encouraged to follow the suggested authoring process. In addition to the task description, participants were also given access to all the domain model components of LBITS [2], a tutoring system for English language skills. They were also provided with an ontology for database modelling as an example. Finally, the students were instructed to use a particular interface style (one free-form text box per fraction) when building their tutor. The participants were allocated a period of six weeks to complete the task, however, most students only started working on it at the end of week 3.

Twelve out of the 13 participants completed the task satisfactorily, i.e. produced working tutoring systems. One participant failed to complete the final step of converting the pseudo-code constraints to the target constraint language. We only present results of the twelve participants in the rest of the paper.

Analysis of CAS's logs revealed that the participants spent a total of 31.3 (13.4) hours on average interacting with the system. Six and a half hours (4.34) of that was spent interacting with the "ontology view". The majority of the time in the "ontology view" was spent developing ontologies. There was a very high variance in the total interaction time, which can be attributed to each individual's ability.

The participants used the textual editors that were available under the "domain model editor" tab to modify/add domain model components required for WETAS, including problems and their solutions, syntax and semantic constraints and macros. The participants spent a mean total of 24.7 (9.6) hours interacting with the textual editors.

	Constraints		*Coverage*		*Completeness*	
	Syntax	Semantic	Syntax (8)	Semantic (13)	Syntax	Semantic
S1	5	12	5	7	63%	54%
S2	5	13	5	12	63%	92%
S3	4	12	4	12	50%	92%
S4	16	16	5	12	63%	92%
S5	14	18	8	13	100%	100%
S6	15	11	5	12	63%	92%
S7	2	5	3	3	38%	23%
S8	8	13	7	4	88%	31%
S9	5	8	4	4	50%	31%
S10	7	11	5	12	63%	92%
S11	4	18	5	12	63%	92%
S12	9	16	6	1	75%	8%
Mean	7.83	12.75	5.17	8.67	64.58%	66.67%
S.D.	4.73	3.89	1.34	4.5	16.71%	34.61%

Table 1: Total Numbers of Constraints Composed by Participants

In order to evaluate the completeness of participants' constraint bases, we manually compiled an "ideal" set of constraints for the domain, containing eight syntax constraints and 13 semantic constraints. The syntax constraints ensure that each component of a solution (such as the LCD and converted fractions) is in the correct format, and the correct problem-solving procedure is followed. The semantic constraints ensure that each component of a solution is correctly defined.

Table 1 lists the total numbers of constraints (syntax and semantic) composed by the participants under the "Constraints" column. The total numbers of "ideal"

constraints covered by each constraint base are listed under the “Coverage” column. The completeness of each constraint base, calculated as the percentage of constraints accounted for by each constraint base, is given under the “Completeness” column. The participants accounted for five syntax constraints in the “ideal” set (65%) and nine semantic constraints (66%). Only one participant (S5) produced all the necessary constraints. The majority of the others had accounted for over half of the desired constraints. One participant (S12) struggled with composing semantic constraints and only managed to account for one desired constraint.

The “ideal” set of constraints contains five syntax constraints for verifying the syntactic validity of inputs, such as whether the LCD is an integer and whether the entered fractions are syntactically valid. They need to be verified due to the generic nature of the interface the students were told to use, which consisted of a single input box to input a fraction. For example, constraints are required to verify that fractions are of the ``numerator / denominator" form. However, these constraints are redundant for the solution-composition interface produced by CAS from a complete ontology. It contains two text boxes for inputting a fraction (one for its numerator and the other for its denominator), ensuring that a fraction is of the correct format. Furthermore, these input boxes only accept values of the type specified for the relevant property in the ontology (numerator and denominator would both be defined as integer in this case). As a consequence, constraints such as the ones to verify whether the specified numerator is an integer are also redundant.

	Constraints		Coverage		Completeness	
	Syntax	Semantic	Syntax (3)	Semantic (13)	Syntax	Semantic
S1	7	15	3	12	100%	92%
S2	8	12	3	12	100%	92%
S3	18	0	3	0	100%	
S4	6	23	3	1	100%	8%
S5	9	8	3	12	100%	92%
S6	0	23	0	1		8%
S7	13	26	3	1	100%	8%
S8	6	0	3	0	100%	
S9	11	23	3	1	100%	8%
S10	9	12	3	12	100%	92%
S11	7	12	3	12	100%	92%
S12	17	42	2	1	67%	8%
Mean	9.25	16.33	2.67	5.42	97.00%	49.97%
S.D.	4.97	11.86	0.89	5.82	9.95%	44.30%

Table 2: Total Numbers of Constraints Generated by CAS

The participants were free to make any modifications to the initial set of constraints produced by CAS, including deleting all of them and composing a new set manually. For that reason, we also analysed the constraints produced by CAS automatically, from the domain information supplied by the author. The generated constraint sets were analysed to calculate their completeness (Table 2). CAS generated the three syntax constraints necessary for CAS’s solution interface for 10 participants. There was one situation where just two required constraints were generated (S12), as the result of an incorrectly specified solution structure. The syntax constraints generator failed produce any constraints for participant S6 due to a bug.

CAS only has the ability to generate 12 out of the 13 required semantic constraints, as it is unable to generate constraints that require algebraic functionality. CAS cannot generate a constraint that accounts for common multiples of the two denominators

larger than the lowest common multiple. So the maximum degree of completeness that can be expected is 92%.

CAS generated the maximum possible 12 semantic constraints from domain-dependant components supplied by five participants. However, it was not successful in generating constraints for the remaining participants. Further analysis revealed that there were two main reasons. One of the reasons was that two of the participants (S4 and S9) had unknowingly added empty duplicate solutions for problems, which resulted in constraints that allowed empty solutions. This situation can be avoided easily by restricting the solution interface not to save empty solutions.

Another common cause for not generating useful semantic constraints was an incomplete ontology. Four participants (S4, S6, S7 and S12) modelled the *Fraction* concept with only a single property of type *String*. This results in a set of constraints that compare each component of the student solution against the respective ideal solution component as a whole. These constraints are not of the correct level of granularity. Consequently, the resulting feedback is limited in pedagogical significance. For example, the constraints would have the ability to denote that the student has made an error in the sum, but not able to pinpoint whether the mistake is in the numerator or the denominator. We believe that the decision to model the *Fraction* concept with a single property may have been influenced by the student interface that we required them to use. The participants may have attempted to produce an ontology and solution structure that is consistent with this particular student interface.

The constraint generator failed to produce any semantic constraints for two participants, S3 and S8. It failed to generate constraints for S3 due to a bug in the system. The other participant (S8) did not add any solutions, and therefore semantic constraints could not be generated.

```
a. Relevance:     Fraction-1 component of IS has a (?var1, ?*) 'Improper fraction'
   Satisfaction:  Fraction-2 component of SS has a (?var1, ?*) 'Improper fraction'
b. Relevance:     (match IS Fraction-1 ("2." ?var1 ?IS-var2 ?*))
   Satisfaction:  (match SS Fraction-1 ("2." ?var1 ?SS-var2 ?*))
```

Figure 2: An example of translating a pseudo-code constraint into WETAS language

Although we assumed that the generated high-level constraints assisted the participants, there was little evidence in their reports that supported this assumption. Only one participant indicated that the generated constraints assisted him. Since no explanation on the high-level constraint representation was provided to the participants, they may have struggled to understand the notation and to find commonalities between the two representations. For example, Figure 2a shows the pseudo-code representation of a constraint that ensures that the numerator of first fraction specified by the student (SS) is the same as the corresponding value in the ideal solution (IS). The equivalent constraint in the WETAS language is given in Figure 2b.

3. Discussion

Analysing the results from the evaluation study confirmed all our hypotheses. Our first hypothesis about CAS being effective has been confirmed in previous work [7, 9]. The 2006 study revealed that CAS was able to generate all the required syntax constraints for 10 of the 12 participants. Furthermore, CAS generated over 90% of the semantic constraints for half of the participants. Considering that the participants were given

very little training in using the authoring system, the results are very encouraging. Providing the users with more training and improving CAS to be fully integrated with a tutoring server (similar to WETAS) would further increase its effectiveness.

The second hypothesis claims that CAS requires less effort than composing constraints manually. In order to obtain a measure for the effort required for producing constraints using CAS, we calculated the average time required for producing a single constraint. Only the participants whose domain model components resulted in generating near complete constraint bases were used for calculating the average effort, to ensure that incorrectly generated constraints were not accounted. Five participants (S1, S2, S5, S10, S11) spent a total of 24.8 hours composing the required domain-dependant information. They also spent a total of 115.04 hours interacting with the textual editors to produce a total of 107 constraints. Consequently, the participants spent a total of 1.3 hours on average to produce one constraint.

The average time of 1.3 hours per constraint is very close to the 1.1 hours per constraint reported by Mitrovic [4] for composing constraints for SQL-Tutor. The time estimated by Mitrovic can be considered as biased since she is an expert of SQL and knowledge engineering, in addition to being an expert in composing constraints. Therefore, the achievement by novice ITS authors producing constraints in a time similar to the time reported by Mitrovic is significant. Furthermore, the time of 1.3 hours is a significant improvement from the two hours required by a similar study reported in [8]. Although that study involved composing constraints for the domain of adjectives in the English language, the overall complexities of the tasks are similar. Further, after the experiment was completed, it was discovered that the setup of WETAS was not optimal, which led the participants to perform additional effort when writing the final constraints. The figure of 1.3 hours per constraint is therefore probably pessimistic.

Although CAS currently generates constraints in a high-level language, it can be extended to generate constraints directly in the language required for execution. Assuming the generated constraints were produced in the required runnable form, the total of 99 syntax and semantic constraints were produced from 24.8 hours spent on composing the required domain information. Consequently, the participants would only require an average of 15 minutes (0.25 hours) to generate one constraint.

An average of 15 minutes to produce a constraint is a significant improvement from 1.1 hours reported by Mitrovic [4]. The time is more significant as the authoring process was driven by novice ITS authors. However, this does not take into account validating the generated constraints. As the constraint generation algorithm may not produce all the required constraints, the domain author may also be required to modify the generated constraints or add new constraints manually.

Finally, the third hypothesis concentrates on the sensitivity of the constraint generation on the quality of the ontology. Developing a domain ontology is a design task, which depends on the creator's perceptions of the domain. The ontologies developed by two users, especially if the domain is complicated, are very likely to be different. The constraint learning algorithm generated constraints using ontologies developed by the 12 participants. Although the ontologies were different, the syntax constraint generation algorithm managed to produce full constraint sets for almost all participants. However, the semantic constraint generation was more sensitive to the ontology. In particular, it was reliant on defining the *Fraction* concept with enough details. The semantic constraint generator managed to produce 92% complete constraint sets of the correct granularity for ontologies with a correctly defined

Fraction concept. On the contrary, the constraints generated for ontologies with a partially defined *Fraction* concept were too general. They compared each fraction composed by students as a whole against its corresponding fraction in the ideal solution. These constraints result in feedback limited in pedagogical significance.

4. Conclusions

This paper reports on an evaluation study of CAS, an authoring system for constraint-based tutors. The evaluation study conducted with novice authors produced very encouraging results. The syntax constraint generator was extremely effective, producing complete constraint sets for almost every participant. The semantic constraints generator produced constraint sets that were over 90% accurate for half of the participants. Although the constraints produced by the generator for the remaining participants were too general, ITS authors can modify them with little effort to produce constraints of correct granularity. We believe that this would contribute towards the reduction of the author's total workload.

At the time of the evaluation, constraints produced by CAS were not runnable, but the system can be easily extended to produce constraints in the target language. This would dramatically reduce the effort required for composing constraints. The evaluation revealed that the total workload required by a novice to generate a single constraint using CAS would be even less than the time required by experts to write them by hand.

We intend to enhance CAS further to generate constraints directly in the runnable form. CAS should also be fully integrated with WETAS and made consistent with its internal representation. We also plan to conduct further evaluations on the effectiveness of CAS's constraint generation.

References

1. Aleven, V., McLaren, B., Sewall, J. and Koedinger, K., The Cognitive Tutor Authoring Tools (CTAT): Preliminary Evaluation of Efficiency Gains. in *ITS 2006*, (Taiwan, 2006), Springer-Verlag, 61-70.
2. Martin, B. and Mitrovic, A., Domain Modeling: Art or Science? in *AIED 2003*, (Sydney, Australia, 2003), IOS Press, 183-190.
3. Martin, B. and Mitrovic, A., WETAS: a Web-Based Authoring System for Constraint-Based ITS. in *AH 2002*, (Malaga, Spain, 2002), LCNS, 543-546.
4. Mitrovic, A., Koedinger, K. and Martin, B., A comparative analysis of cognitive tutoring and constraint-based modeling. in *UM 2003*, (Pittsburgh, USA, 2003), Springer-Verlag, 313-322.
5. Mitrovic, A., Mayo, M., Suraweera, P. and Martin, B., Constraint-based Tutors: a Success Story. in *14th Int. Conf on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*, (Budapest, 2001), Springer-Verlag, 931-940.
6. Murray, T. Expanding the Knowledge Acquisition Bottleneck for Intelligent Tutoring Systems. *International Journal of Artificial Intelligence in Education*, *8*. 222-232.
7. Suraweera, P., Mitrovic, A. and Martin, B., A Knowledge Acquisition System for Constraint-based Intelligent Tutoring Systems. in *AIED 2005*, (Amsterdam, Netherlands, 2005), IOS Press, 638-645.
8. Suraweera, P., Mitrovic, A. and Martin, B., The role of domain ontology in knowledge acquisition for ITSs. in *ITS 2004*, (Maceio, Brazil, 2004), Springer, 207-216.
9. Suraweera, P., Mitrovic, A. and Martin, B., The use of ontologies in ITS domain knowledge authoring. in *SWEL '04, ITS 2004*, (Maceio, Brazil, 2004), 41-49.
10. Tecuci, G. *Building Intelligent Agents: An Apprenticeship Multistrategy Learning Theory, Methodology, Tool and Case Studies*. Academic press, 1998.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Taking advantage of the Semantics of a Lesson Graph based on Learning Objects

Olivier MOTELET [a,1], Nelson BALOIAN [a], Benjamin PIWOWARSKI [b], José A. PINO [a]
[a] *Universidad de Chile*
[b] *Yahoo! Research Latin America*

Abstract. Lesson graphs are composed of Learning Objects (LOs) and include a valuable amount of information about the content and usage of the LOs, described by the LO metadata. Graphs also make explicit the links between the LOs (i.e. the graph semantics). This article proposes a conceptual model for taking advantage of this information. This model is based on an original diffusion process that copes with the problem of lesson graphs where some metadatas are missing. Two applications of the model are described and were implemented over a previously developed lesson authoring tool whose goal is to facilitate the lesson authoring process.

Keywords. Learning Object Metadata, Lesson Graph, Graph Semantics

1. Introduction

The last ten years have witnessed the emergence of the concept of Learning Objects (LO) and learning object repositories (LOR). Although LOs have received several definitions in the literature, in this article we will simply consider a LO as a piece of multimedia educational material (a slide, a web page, a simulation, etc.) and its corresponding metadata. One of the main ongoing efforts in this area is the specification of a standard for the metadata characterizing a LO, the so-called Learning Object Metadata (LOM) [7]. Compared with other metadata standards for documents, that mainly address physical attributes of the digital resources, LOM offers a large set of educational attributes, such as the difficulty, the interactivity degree and the description of the pedagogical goals of the LO. This model motivated the development of various systems processing metadata in order to ease LO authoring [5], LO use [2], and LO retrieval [13]. This paper addresses the case of a teacher engaged in a lesson authoring process where the lesson does not consist of a single LO but a graph including many fine-grained LOs. Each node of the graph is a LO and each edge denotes a semantical or rhetorical relation between two LOs. We present a conceptual model for building systems taking advantage of the lesson graph semantics. Two different applications of this model are described and used in order to help the author performing the three tasks: a) defining the context of a query within a

[1]Correspondence to: Olivier Motelet, DCC - Universidad de Chile, Avenida Blanco Encalada 2120, Tercer Piso, Santiago, Chile , C.P. 837-0459, Tel (+56 2) 678.4365, E-mail: omotelet@dcc.uchile.cl

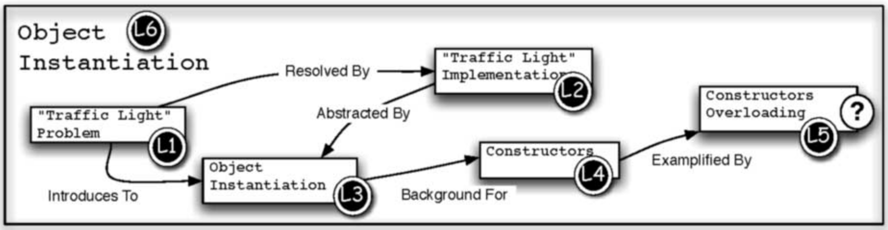


Figure 1. Start of a lesson graph about "object instantiation"

LO repository in order to enhance retrieval precision, b) setting the metadata values of a given LO, and c) checking the appropriateness of the metadata values the user set.

The next section describes the LOM specification and defines the concept of lesson graph. Then, a conceptual model for taking advantage of the graph semantics is presented along with two applications. Next, the potential of these applications to support the lesson authoring process is introduced. Finally, model limitations and outlook are discussed.

2. Metadata and Lesson Graph

LOM has about 60 attributes describing technical, educational and general aspects of educational resources. Attributes are identified by a series of names separated by slashes, e.g. `general/title`, where "general" is the category and "title" the attribute name. Attributes can be classified in three groups: (1) Predefined vocabulary values (e.g. *easy* and *difficult* are vocabulary values for the `educational/difficulty` attribute). (2) Free text. (3) Primitive types, e.g. identifier, date, time, or integer. A range is defined for most attribute values like e.g. a set of strings for `general/keywords`.

Many authors have chosen the graph as the most suitable way of structuring the learning material of computer-based learning systems whenever adaptability and flexibility of the learning material is required [9,3]. In a LOM-based lesson graph, the `relation` attribute of LOM is used to describe the links between the LOs of the lesson. Links are typed, e.g., *introducesTo*, *isPartOf*, or *exemplifiedBy*. The set of links defines the edges of a lesson graph in which the nodes are the LOs. Such a graph is called **LO graph**. Figure 1 illustrates a LO graph consisting in LOs and relations among them. Six LOs, labeled from `L1` to `L6` describe a part of a programming course of an object oriented language. `L1` describes the problem (how coordinate traffic lights in a crossroad) and `L2` presents its implementation as a Java program. This problem aims to teach object instantiation in a program. `L3` and `L4` refer to documents defining object instantiation and the concept of constructors respectively. `L5` is a node inside the lesson graph whose learning material has still not been defined. `L6` is a LO of coarser granularity and acts as a container for `L1` to `L5`.

Most used LOM relation types were often inspired by the DublinCore [1] specification. However, these relations were not originally designed for educational authoring, so they are not well suited to cope with the requirements of lesson authoring. We opted for another extended taxonomy proposed by Trigg [14]. Trigg's taxonomy defines an extensive set of relations supporting narration, that can be used to define a lesson graph. It defines semantical as well as rhetorical relations allowing the author to use those that

better match the needs of the specific lesson. We asked a group of lecturers working in the Computer Science Department of our university to organize the content of the introductory computer programming course for freshmen as a lesson graph using a certain set of relations. We specifically left out the too generic relations (e.g. *isFollowedBy*) since they give almost no information about a LO context in the graph. Based on their work, we empirically selected a subset of these relations, emphasizing the semantical, rhetorical and organizational aspects of course authoring: *introducesTo, assessedBy, supportedBy, abstractedBy, exemplifiedBy, comparableWith, backgroundFor, summarizedBy, resolvedBy, isPartOf*. Each of these relations has an opposite: a relation from a LO *a* to another LO *b* implies an opposite relation between *b* and *a* (e.g. *isPartof* defines a reverse relation *hasPart*). In this article, lesson graphs are built using this set.

3. Taking advantage of the Graph Semantics

In the literature, the dependency between graph semantics and the metadata values of graph LOs has been explored and we can distinguish two approaches: (1) Using metadata semantic to influence graph semantics. Farrell et al. [2] uses this approach in order to dynamically assemble repository LOs. (2) Using graph semantics to influence metadata semantics. Hatala and Richards [5] uses this method in order to suggest values for missing LOM elements. This article focuses on this second approach.

We call **influence rules** the rules defining the influence of the graph semantics on the metadata semantics. For instance, the rule of [5], *When there is a parent-child (i.e.* isPartOf*) relation between two LOs, the value of the* `educational/intentedUserRole` *attribute of the parent may be strongly suggested to the child*, is an influence rule . The result of this rule is a possible metadata value that may be "strongly" suggested to characterize the child LO. We call **contextualized metadata value (CMV)** the result of computing an influence rule over the **metadata values of a LO graph**.

In [5], the influence rules makes use of the existing metadata values in order to generate CMVs. When some metadata values are missing, this process is directly affected since there may be no input for the rules. Considering that various studies witness that numerous metadata values are missing in the available LOs [4], this could be a serious issue. In order to cope with this problem, we propose a diffusion process in which for a given LO, each influence rule is applied *not only* to the metadata values of the neighboring LOs *but also* to the CMVs previously generated by this rule or other ones. Since this process increases the input scope of the influence rules, the impact of missing metadata value when computing the rules should decrease but at the price of the possible generation of noise, that is unwanted metadata values. We call this recursive process the **context diffusion**.

Figure 2 depicts a conceptual model for generating CMVs using context diffusion. In this model, existing metadata values for a LO graph are processed by a set of influence rules in order to generate CMVs. In contrast with existing approaches for which rule computation is directly done on the metadata values of the graph LOs, in this model, rule processing depends on our context diffusion mechanism.

During context diffusion, influence rules are iteratively applied on both CMVs and original metadata values of the graph until the generated CMVs finally converge. Defining such a non-monotonic process over an existing inference rule system for the seman-

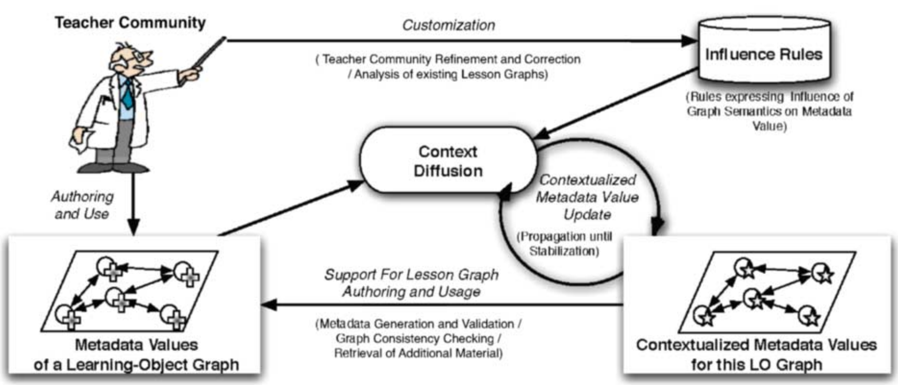


Figure 2. Conceptual model for taking advantage of the lesson graph semantics

tic web such as Jena [6] is a non-trivial task. Therefore, we use a simple push protocol based on propagation: For a certain node, the original metadata values and CMVs of the neighboring nodes are combined with its current CMVs using the influence rules. This update process generates a new set of CMVs for this node. If it appears that the node's CMVs changed during this process, all its neighbors are also updated in turn. Otherwise, the diffusion process stops for that node. Since we consider a lesson graph as a cyclic graph with symmetric edges (see Section 2), update computation should be chosen in such a way that it guarantees the diffusion process convergence.

Teacher communities are also part of the model : (1) As they author and use the lesson graphs they benefit from the support provided by using the CMVs, (2) In order to adapt the system to their teaching style and preferences, they can customize the influence rules manually (e.g. defining new influence rules or refining the existing ones) or automatically (e.g. providing existing lesson graphs that the system may analyze in order to deduce the necessary information for customizing the influence rules).

Instantiating the conceptual model presented in this section consists in defining (1) the nature of the CMVs, (2) the influence rules with a restricted language relating graph and metadata semantics, and (3) a CMVs update process that guarantees convergence. Two applications of the model are described in the next section.

4. Applications of the conceptual Model

This section presents two uses of the above described model. In the first one, the graph consistency is analyzed generating restrictions for some of the metadata values. In the second one, similarities among the attribute values of the graph's LOs are used to generate suggestions for the metadata value of the LOs. In this section, we describe for both examples how we defined the influence rules, the CMVs, and the update process.

4.1. Attribute Similarities and Value Suggestion

As stated by Hatala et al. [5], graph semantic analysis may be used to identify similarities between the metadata attribute values of related LOs. In the graph of Figure 1, since `L1` *introduces* `L3` we may expect that the attribute `general/keyword` of `L1` and `L3` share

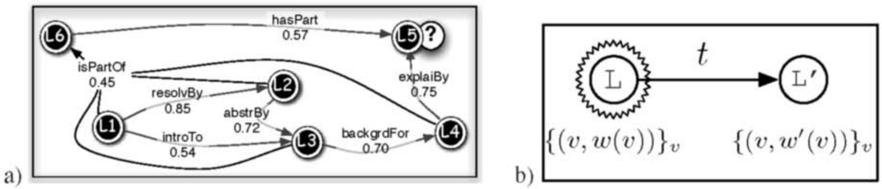


Figure 3. (a) Weights of suggestions for `general/keyword` attribute. (b) Update process of suggestions.

some values. In the example, the values of this attribute are *{instantiation,object,method}* for `L1` and *{instantiation,object,new}* for `L3`, sharing two common values out of three.

Rule Definition. In order to extend the predefined set of rules of [5], we propose to weight similarities for each couple of attribute and relation type by analyzing an existing repository of lesson graphs (containing about 170 LOs) developed in our institution. We found out that LOs linked with a *introducesTo* relation share the same values for the `general/keyword` attribute with a probability 0.54. Figure 3a is a reduced view of the graph of Figure 1 showing the probability (calculated on our corpus) of sharing the same values between neighboring nodes for the `general/keyword` attribute. Influence rules are defined as a triplet consisting of metadata attribute, relation type, and similarity probability.

CMV Definition. The previous rules are used to generate a list of special CMVs called **suggestions** for each LOs of a lesson graph. A suggestion is a set $\{v, w(v)\}_v$ associating a weight $w(v)$ to all the possible values v for a certain attribute a of a LO `L`. The weight is 0 when the value is not at all appropriate for `L`, while it is 1 when it fits it perfectly. At the beginning of the context diffusion process, we set $w(v) = 0$ for all possible values v for the a attribute except for the original values which have a weight 1.

Update Process. Figure 3b depicts a diffusion step: for an attribute a, changes in the suggestion $\{(v, w(v))\}_v$ of a LO `L` are propagated in order to influence the suggestion $\{(v, w'(v))\}_v$ of every other LO `L′` connected to `L`. We note $p_a(t)$ the probability that `L` and `L′` share the same value for the a attribute giving that a relationship of type t connects `L` with `L′`. The update process consists of replacing the CMV of `L′` with $\{(v, max(w'(v), p_a(t) \times w(v))\}_v$.

In cases where the same value for a certain metadata attribute is suggested by more than one neighboring LO, the maximum weight is considered. Since on the one hand, the value weights are filtered with the operator maximum and on the other hand, these weights are decreased at each propagation step (since they are multiplied by a probability between 0 and 1), we can guarantee that the process converges.

4.2. Graph Consistency and Value Restriction

Let us consider again the lesson graph of Figure 1. It is plausible to think that for a certain teacher community, the fact that `L1` introduces `L3` may imply that the content of the `L1` LO is simpler than the one of `L3` (otherwise the lesson graph would not be consistent). In terms of LOM semantics, it means the value of the LOM attribute `educational/difficulty` of `L1` should be lower or equal to the `educational/difficulty` of `L3`. If `L1` introduces to more than one LO, its level of `educational/difficulty` should be compatible (lower) than each element it introduces

a)

	LOM Attributes				
	General		Educational		
Relation Types	Coverage	Aggregation Level	Semantic Density	Typical Learning Time	Difficulty
isPartOf	$\subseteq \bigcap v_i$	$\leq \min v_i$		$\leq \max v_i$	
hasPart	$\supseteq \bigcup v_i$	$\geq \max v_i$			$\geq \min v_i$
summarizes		$\leq \min v_i$	$\leq \min v_i$	$\leq \max v_i$	$\leq \max v_i$
summarizedby	$\supseteq \bigcup v_i$	$\geq \max v_i$	$\geq \max v_i$		
introducesTo		$\leq \min v_i$	$\leq \min v_i$	$\leq \max v_i$	$\leq \max v_i$
introducedBy		$\geq \max v_i$	$\geq \max v_i$		
assessedBy		$\geq \max v_i$			
assesses		$\leq \min v_i$		$\leq \max v_i$	$\leq \max v_i$

b)

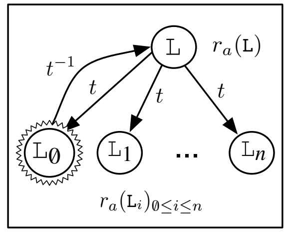


Figure 4. (a) Examples of restriction rules. (b) Update process of restriction intervals.

to. Since the value of `educational/difficulty` is associated to a predefined vocabulary, we can define an order between the terms of this vocabulary to test the consistency.

Rule Definition. The previous assumption about the consistency of LOM attribute values is a special type of influence rule called a **restriction rule**. A restriction rule is defined as the combination of three elements: (1) a metadata attribute name, (2) a relation name, (3) a couple of operators of the set $\{\leq, \geq\} \times \{\mathtt{max}, \mathtt{min}\}$ (or $\{\subseteq, \supseteq\} \times \{\cup, \cap\}$ when dealing with a metadata attribute based on a set of elements). For example, the restriction rule for the attribute `educational/difficulty` and the relation *introducesTo* is defined as $\leq \mathtt{max}\ v_i$ where v_i is an attribute value of a LO related with the *introducesTo* connection. Other examples of rules can be found in Figure 4a. Note that the rules can be modified in order to adapt them to other educational contexts and new attributes.

CMV Definition. Applying restriction rules to the LO graph results in a special CMV called **restriction interval** for each metadata attribute of each LO. We define the restriction $r_a(\mathtt{L})$ for the LO L and the attribute a as the interval $[r_a^{lower}(\mathtt{L}), r_a^{upper}(\mathtt{L})]$. The possible values of a lie within this interval. We can also detect anomalies when the value of a set by the user does not belong to it. At the beginning of the diffusion process, the restriction interval of the a attribute of a LO L is initialized to the whole interval of the possible values ($[a_{min}, a_{max}]$) unless it has been set by the user. In the latter case, the interval reduces to $[a(\mathtt{L}), a(\mathtt{L})]$.

Update Process. Figure 4b depicts a step of the diffusion process: the changes in the restriction interval for a certain attribute a of a LO $\mathtt{L}_\emptyset$ are propagated to the restriction interval of its neighbor L. Restriction rules are applied to the reverse of the relation type used for propagating the changes: in this case, the relation of type t connecting the LO L (receiving the update notification) to $\mathtt{L}_\emptyset$ (propagating changes) is used. The update process of a restriction rule considers the $n+1$ LOs to which L is connected with relations of type t. Those LOs are noted $\mathtt{L}_i$ with $\emptyset \leq i \leq n$ and $n \in \mathbf{N}$.

If the relation of type t imposes a restriction rule $\leq \mathtt{max}$ for the values of the attribute a, the update process consists in replacing the restriction interval $r_a(\mathtt{L})$ of the LO L by:

$$r_a(\mathtt{L}) \cap \bigcup_i \left] -\infty, r_a^{upper}(\mathtt{L}_i) \right]$$

This means that the restriction interval of L is intersected with an interval consisting of an infinite lower boundary (no effect on $r_a(\mathtt{L})$) and an upper boundary equal to the maximum of the upper boundaries of the restriction intervals of the $\mathtt{L}_i$ LOs (may lower the upper boundary of $r_a(\mathtt{L})$). The update processes for restriction rules of type $\geq \mathtt{max}$,

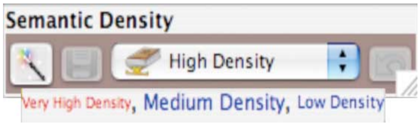


Figure 5. Weighted values for the `Educational/SemanticDensity` attribute. Note that the value `Very High Density` is differently painted showing that it does not enter the scope of the restrictions for this attribute.

$\leq$ `min`, or $\geq$ `min` follow similar principles. Note that the diffusion converges since it is only based on monotonically slimming down restriction intervals. If a restriction interval becomes void during the diffusion process, it means that an incoherency occurred in the graph. This incoherency could be due to (a) an incorrect value for the metadata of the LO, (b) some incorrect relations between this LO and the other LOs of the graph, or (c) a contradiction between two restrictions rules that should be resolved. Diffusion follows the same scheme when dealing with metadata attributes based on value sets.

5. Model Implementation in a Lesson Authoring Tool

The two applications of the model presented above were implemented in LessonMapper2, a Java-software prototype for authoring lesson graphs of LOs characterized with LOM. In this tool, our model is used to facilitate lesson authoring.

Suggestions and restrictions are used to alleviate the metadata generation process. When a lesson author is setting a metadata attribute, a weighted list of possible values is displayed: Suggested values with higher associated weight are displayed with the biggest font as shown in Figure 5. Suggestions not complying with the restrictions are highlighted. Restrictions are also used for detecting incoherences in the graph: all metadata values of the graph LOs are constantly checked. Whenever there is a restriction conflict, the corresponding attribute values are highlighted on the lesson graph. More details about this feature can be found in [11].

Suggestions and restrictions computed for an empty node in a lesson graph (like `L5` in Figure 1) can be used to refine a query on a learning object repository. When querying a repository, searching for a LO to fit into the empty node, the results are checked with respect to the restrictions generated by the system: Results complying with all restrictions are better ranked. Suggestions were used for enhancing retrieval: when querying a learning object repository containing other lesson graphs, the suggestions for the empty node are compared with the suggestions of the proposed results. The results having suggestions similar to the ones of the empty node are better ranked. Ranking information is then combined with a classical keyword-based query processed by the information retrieval system Lucene [8]. Small-scale experiments have shown that this approach effectively enhances the retrieval of LOs compared with Lucene alone [12].

6. Conclusion

This article presented a conceptual model using a context diffusion process in order to take advantage of the semantics of a lesson graph based on LOs and their metadata. Two illustrative applications were described: one dealing with the extension of an existing

method for generating metadata value suggestions and the other one dealing with the lesson graph consistency. We also showed how these features were implemented within a tool designed for lesson authoring.

Depending on the lesson graph size, the proposed context diffusion process may decrease the effect of missing metadata values compared with the other existing methods. For instance, since the influence rules for value suggestions are automatically defined for all the couples relation type - attribute, context diffusion extends their scope to all the available values of the graph.

While our system needs at least some metadata value, LOM usage during lesson authoring is generally limited to enable the sharing process. In [11], LOM are used to visually characterize the lesson graph elements, but the effects of this proposal are not evaluated. Moreover, our system requires the lesson author to define explicit relations. Whereas some studies argue for the benefits of the graph structure for building lesson [10], the cognitive weight and the advantage of typed relations on the lesson authoring process remains to be measured. Nevertheless, metadata types and relation types are parameters of our context diffusion process: Adapting them to other specific needs only impacts the influence rules not the diffusion process. For the same reason, while this article has focused on graph-based lesson authoring, our model could be applied to other settings using metadata-based graphs.

References

[1] DublinCore. Metadata initiative. `http://www.dublincore.org` last visit on 05/2006, 2007.

[2] R. Farrell, S. D. Liburd, and J. C. Thomas. Dynamic assembly of learning objects. In *World-Wide Web International Conference WWW 2004, New York*, 2004.

[3] S. Fischer. Course and exercise sequencing using metadata in adaptive hypermedia learning systems. *J. Educ. Resour. Comput.*, 1(1es):5, 2001.

[4] N. Friesen. International lom survey report. Technical report, ISO/IEC JTC1/SC36, 2004.

[5] M. Hatala and G. Richards. Value-added metatagging: Ontology and rule based methods for smarter metadata. In LNCS, editor, *Conference on Rules and Rule Markup Languages for the Semantic Web RuleML*, volume 2876 of *Lecture Notes in Computer Science*, pages 65–80. Springer, 2003.

[6] JENA. A semantic web framework for java. `http://jena.sourceforge.net/`, 2007.

[7] LOM. Ieee ltsc p1484.12.1 learning object metadata specification final draft. `http://ieeeltsc.org/wg12LOM/`, 2002.

[8] Lucene. Full-featured text search engine library in java. `http://lucene.apache.org`, 2007.

[9] G. McCalla. *Foundations and Frontiers of Adaptive Learning Environments*, chapter The search for adaptability, flexibility, and individualization: Approaches to curriculum in intelligent tutoring systems., pages 91–122. Springer, 1992.

[10] E. McDaniel, B. Roth, and M. Miller. Concept mapping as a tool for curriculum design. In *Informing Science IT Education Conference (InSITE2005)*, 2005.

[11] O. Motelet and N. A. Baloian. Hybrid system for generating learning object metadata. In *ICALT*, pages 563–567. IEEE Computer Society, 2006.

[12] O. Motelet, B. Piwowarski, G. Dupret, and N. Baloian. Enhancing educational material retrieval using lesson authoring context. Technical report, DCC - Universidad de Chile, 2006.

[13] N. Pinkwart, M. Jansen, M. Oelinger, L. Korchounova, and U. Hoppe. Partial generation of contextualized metadata in a collaborative modeling environment. In *2nd International Workshop on Applications of Semantic Web Technologies for E-Learning AH 2004, Eindhoven, Netherlands*, 2004.

[14] R. Trigg. *A Network-Based Approach to Text Handling for the Online Scientific Community*. PhD thesis, University of Maryland, November 1983.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Predicting Students' Performance with SimStudent: Learning Cognitive Skills from Observation[1]

Noboru MATSUDA[a], William W. COHEN[b], Jonathan SEWALL[a], Gustavo LACERDA[b] and Kenneth R. KOEDINGER[a]

[a]*Human-Computer Interaction Institute*
[b]*Machine Learning Department*
Carnegie Mellon University

Abstract. SimStudent is a machine-learning agent that learns cognitive skills by demonstration. SimStudent was originally built as a building block for Cognitive Tutor Authoring Tools to help an author build a cognitive model without significant programming. In this paper, we evaluate a second use of SimStudent, *viz.,* student modeling for Intelligent Tutoring Systems. The basic idea is to have SimStudent observe human students solving problems. It then creates a cognitive model that can replicate the students' performance. If the model is accurate, it would predict the human students' performance on novel problems. An evaluation study showed that when trained on 15 problems, SimStudent accurately predicted the human students' correct behavior on the novel problems more than 80% of the time. However, the current implementation of SimStudent does not accurately predict when the human students make errors.

Keywords. Cognitive modeling, programming by demonstration, inductive logic programming, Cognitive Tutor, intelligent authoring.

Introduction

SimStudent [1] was originally built as an intelligent building block for the Cognitive Tutor Authoring Tools (CTAT) [2]. In this authoring environment, an author can build a cognitive model that represents domain principles for a target task without significant AI-programming. Instead, the author is asked only to *demonstrate* the task, both correctly and incorrectly, using a tutor interface built with CTAT. SimStudent then learns how to perform the target task (or how to make a mistake) by generalizing the author's demonstration. The fundamental technology used in SimStudent is *programming by demonstration* [3].

Since SimStudent can generalize actions taken in the tutor interface, theoretically speaking, SimStudent should be able to model the domain principles that a human student acquired while solving problems with the Tutor. In other words, SimStudent can be used to dynamically generate a *student model* for an individual student using Cognitive Tutor.

[1] The research presented in this paper is supported by the National Science Foundation Award No. REC-0537198. This work was also supported in part by the Pittsburgh Science of Learning Center, which is funded by the National Science Foundation Award No. SBE-0354420

The ultimate goal of the current project is to evaluate whether SimStudent can be used for student modeling. A preliminary study showed that SimStudent can model human students' *correct* behaviors [4]. On the other hand, modeling students' erroneous behaviors is quite challenging, partly because the human students make non-systematic errors such as slips and random errors. Although building a descriptive model of student misconceptions (telling why a particular behavior was observed) might be challenging, building a *predictive model* of student performance (telling whether a student would perform a next step correctly or not) might be tractable. Such a predictive model might give us further insight into designing a descriptive model of the student's knowledge, which represents the student's misconceptions. As an initial step towards our ultimate goal, the current paper addresses two research questions: (1) Can SimStudent learn domain principles by observing an individual student's performance? (2) Can SimStudent *predict* an individual student's behavior on novel problems with the domain principles learned from the individual student?

1. Related Studies

There have been a wide variety of machine-learning techniques studied so far for student modeling, including synthetic, analytic, and stochastic methods. Ikeda et al [5] built an inductive logic learner in conjunction with a truth maintenance system to inductively construct a hypothetical student model. Baffes et al [6] applied a machine learning technique to modify a given set of domain principles to be consistent with the student's incorrect behaviors. Similarly, Langley et al [7] applied a theory-refinement technique to model an individual student's behavior given a general model of the domain. MacLaren et al [8] applied a reinforcement learning method based on the ACT-R model [9] to model the process in which the students' knowledge activation patterns are strengthened. There are more studies on applications of machine-learning for student modeling.

The current study is about an application of a cutting-edge technology for student modeling. The proposed system is unique in two ways: (1) the fundamental framework used in SimStudent, inductive logic programming [10], is domain-general, and hence it is applicable to a wide range of domains; (2) SimStudent has a dual role in this context – to automatically model a domain principle used in a Cognitive Tutor, and to model a student's performance while he/she is using the Cognitive Tutor. The former cognitive model enables the Cognitive Tutor to perform model-tracing (see section 2.1), whereas the latter model allows the Cognitive Tutor to diagnose a student's competency to, say, proactively select a problem on which the student is likely to make a mistake, which in turn triggers learning.

2. Experimental Setting: Algebra I Cognitive Tutor

For the current experiment, we used the Algebra I Tutor developed by Carnegie Learning Inc. The Algebra I Tutor is a Cognitive Tutor for high school algebra used in real classroom situations in more than about 2000 schools nationwide in the United States [11].

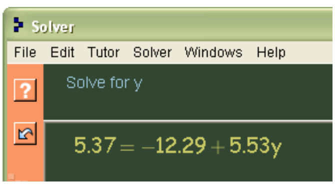
We used tutor-interaction logs representing interactions between the Cognitive Tutor and individual human students. The current data were collected from a study conducted in an urban high school in Pittsburgh. There were 81 students involved in the study.

There were 15 sections taught by the tutor, which covered most of the skills necessary to solve linear equations. In the current study, we focus on the log data for the first eight sections. The equations in those sections only contain a single unknown and have the form A+B=C+D where each of A, B, C, and D is either a constant or an unknown term in the form of N*x* with a constant number N.

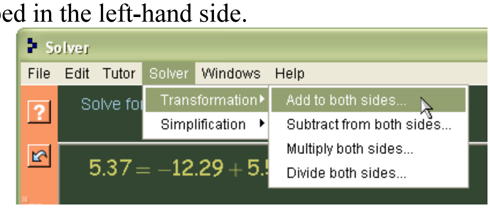
2.1. Tutoring interaction

The unit of analysis in the current paper is a single *problem-solving step* performed by the human students using the Cognitive Tutor interface. A problem-solving step is slightly different from an *equation-transformation step*, which corresponds to an action that transforms one equation into another complete equation when solving an equation with paper and pencil.

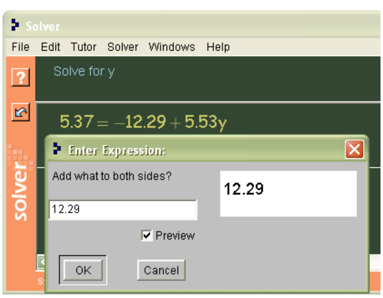
There are two types of problem-solving steps: (1) an *action step* is to select an algebraic operation to transform an equation into another (e.g., to declare that I will "add 3*x* to both sides" of the equation), and (2) a *type-in step* is to do a real arithmetic calculation (e.g., "to enter -4" as a result of adding 3*x* to -4-3*x*). By performing these problem-solving steps, a given equation is transformed as follows: a student first selects an action and then applies it to both sides of the equation. For example, for the equation shown in Figure 1 (a), the student selected "Add to both sides" from the pull down menu (b), which in turn prompts the student to specify a value to add (c). This completes the first problem-solving step, which is an action step. The student then enters the left- and right-hand sides of the new equation separately, in two type-in steps. Figure 1 (d) shows the moment when the student has just typed in the left-hand side.



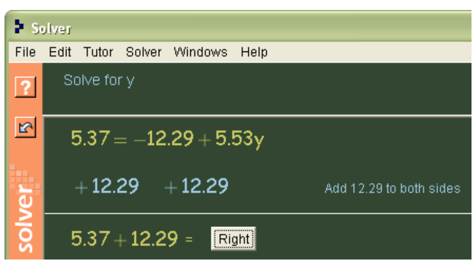
(a) A given equation



(b) Selecting an action



(c) Entering a value to be added



(d) Typing-in a left-hand side

Figure 1: Screen shot for the Algebra I Tutor.

In summary, three problem-solving steps correspond to a single equation-transformation step, which transforms one equation into another. Sometimes, however, the tutor carries out the type-in steps for the student, especially when new skills are introduced.

When a student makes an error, the tutor provides feedback. The student can also ask for a hint (by pressing the "[?]" button on the left side of the tutor window) when he/she gets stuck.

Every time a student performs a step, the tutor logs it. In this study, we are particularly interested in the following log information: (1) The place where the step was made, called *Selection*. This corresponds to an element on the graphical user interface, e.g., the left-hand side of an equation, or a pull-down menu on the menu bar. (2) The name of the skill used, called *Action*, which is either the name of the algebraic operation selected from the menu for an action step, or the symbol "type-in" for a type-in step. (3) The value entered, called *Input*. For example, the value specified (12.29 in Figure 1 (c)) to be added to both sides for the "add" action step mentioned above, or the left- and right-hand side entered for the type-in steps. (4) The correctness of the step, which is either "correct", "error", or "hint" (when the student asked for a hint).

Using the first three pieces of information mentioned above, a problem-solving step is represented with a tuple <Selection, Action, Input>. The tuple plays an important role for SimStudent's learning, as described in section 3.2.

2.2. Cognitive model and model-tracing

As mentioned above, the tutor provides immediate feedback on the steps. This is possible because the tutor has a *cognitive model*, represented as a set of production rules that are sufficient to perform the target task. A cognitive model can include both correct and incorrect (called "buggy") production rules. When a step is performed by a student, the tutor attempts to find a production rule that matches the step. This technique is called *model-tracing*.

3. The Architecture of SimStudent

This section is a brief overview of SimStudent. We first explain how SimStudent learns cognitive skills from demonstrations. The double meaning of "demonstration" in the current context is then explained – a demonstration by an author who is building a Cognitive Tutor, and a "demonstration" in the tutor-interaction log representing a real student's performance. Due to the space limitation, we only provide a brief overview. See Matsuda et al [1] for more details.

3.1. Modeling student-tutor interaction in the Algebra I Tutor

SimStudent learns a single production rule for each problem-solving step demonstrated. In the most general case, when demonstrating a step, the demonstrator must specify two additional things: (1) a skill name, and (2) a focus of attention. In the study using Algebra I Tutor log, however, both of these could be derived without explicit specification as explained below.

The *skill name* must be consistent across problems throughout the demonstration. In the Algebra I Tutor, the skill name for an action step is an abbreviation of the action name shown in the tutor interface. For example, in the step shown in Figure 1 (a-c), the skill name to select "Add 12.29 to both sides" is called "add." The skill name for a type-in step consists of the skill name of the corresponding action step and "-typein." So, for example, the skill name for the type-in step shown in Figure 1 (d) is called "add-typein."

The *focus of attention* is the set of elements on the tutor interface used to determine what action is to be taken. For example, the problem-solving step shown in Figure 1 (b) – "add 12.29 to both sides" – requires two elements, "5.37" and "-12.29+5.53y" as the focus of attention. There is, however, an issue in getting the focus of attention when SimStudent is used to model real students' performance – the real students do not indicate their focus of attention. We have assumed that both the left-hand side and right-hand side are used as the focus of attention for the action steps. For the type-in steps, we presume that the skill name and the expression immediately above the expression to be typed-in are the focus of attention. So, for example, for the type-in step shown in Figure 1 (d), where "5.37+12.29" was typed-in, the skill name "Add 12.29 to both sides" and the expression "5.37" are used as the focus of attention.

3.2. Learning algorithm

Production rules learned by SimStudent are written in the Jess production rule description language [12]. A production rule consists of three major parts: what, when, and how. The what-part specifies which elements of the tutor interface are involved in the production rule. The when-part shows what feature conditions must hold among the elements in the what-part for a production rule to be fired. The how-part specifies what computation should be done with the what-part to generate a correct "Input" (described in section 2.1).

SimStudent utilizes three different learning algorithms to learn these three parts separately. The what-part is learned as a straightforward generalization of the focus of attention. Each element in the focus of attention can be identified uniquely in terms of its location on the tutor interface. Constraints on the location are generalized from the most specific description using absolute positions (e.g., the *2nd* and the *3rd* cells) to the most general description that represents arbitrary elements (e.g., *any two* cells). A moderate generalization uses relative locations (e.g., *two consecutive* cells).

SimStudent uses FOIL [13] to learn the when-part. The target concept is the "applicability" of a particular skill given a focus of attention. When a step for a particular skill S is demonstrated, the instance of demonstration serves as a positive example for the skill S and a negative example for all other skills. We provide FOIL with a set of feature predicates as the background knowledge with which to compose hypotheses for the target concept. Some examples of such feature predicates are `isPolynomial`(_), `isNumeratorOf`(_,_), `isConstant`(_). Once a hypothesis is found for the target concept, the body of the hypothesis becomes the what-part on the left-hand side of the production rule. For example, suppose that FOIL found a hypothesis S(A, B) :- `isPolynomial`(A), `isConstant`(B) for the applicability of the skill S. The left-hand feature condition for this production rule says that "the value specified in the first focus of attention must be polynomial and the second value must be a constant."

SimStudent uses iterative-deepening depth-first search to learn an operator sequence for the right-hand side of the production rules. When a new instance of a particular skill is demonstrated, SimStudent searches for the shortest operator sequence that derives the demonstrated action (i.e., the "Input") from the focus of attention for all instances demonstrated thus far. The operators are provided as background knowledge.

4. Modeling Real Students

How well does SimStudent predict real students' performance? How often does SimStudent make incorrect predictions, whether reporting a correct step as incorrect or an incorrect step as correct? To answer these questions, we analyzed SimStudent's fidelity at predicting human students' performance. We are interested not only in how well SimStudent predicts real students' correct steps, but also – or even more, for educational purposes – in how well it predicts incorrect steps.

4.1. Data: Tutor-interaction log

The students' learning log data were converted into *problem files*. Each problem file contains the sequence of problem-solving steps made by a single real student for a single equation problem. There are three types of steps: *correct*, *buggy*, and *error*. The correct steps are those that were model-traced with a correct production rule by the Carnegie Learning Tutor. The buggy steps are those that were model-traced with a buggy production rule. The error steps were not model-traced either with a correct or a buggy production rule.

There were 1897 problems solved by 81 individual human students. There were a total of 32709 problem-solving steps. These steps contain 21794 (66.7%) correct steps, 2336 (7.1%) buggy steps, 4567 (14.0%) error steps, and 4012 (12.3%) hint seeking steps[2]. Since the Algebra I Tutor already has a wide variety of buggy rules based on empirical studies, the buggy steps likely capture a fair proportion of the incorrect steps that the human students are likely to make. Furthermore, we have found (by manually verifying data) that most of the error steps are likely to be due to slips and random errors in using the tutor interface, which makes the error steps most challenging to be modeled. Thus, we only used correct and buggy steps for the current evaluation.

4.2. Method: Validation

For each individual human student, we have taken the first 15 problems for training and the following five problems for testing. Using these problems, the validation was conducted for *each individual human student* separately as follows:

```
For each of the 15 training problems
    Train SimStudent on the correct and buggy steps  /* learning */
    For each of the five test problems
        Attempt to perform the correct steps  /* validation */
```

Production rules had been updated each time a five-problem validation test was taken place. Notice that only the steps correctly performed by the real student were used for validation. This is because we are evaluating whether SimStudent could correctly *predict* if a real student would successfully perform a step or not. A student's performance on a step is coded as "success" if the *first attempt* at the step is correct; otherwise it is coded as "error." The Cognitive Tutor forced real students to perform each step repeatedly until correct in order to proceed to the next step. Therefore, a chain

[2] Asking for a hint is logged as a problem-solving step. As mentioned in section 4.3.2, whenever a real student asked for a hint on a step, the student's attempt was coded as "error."

of correct steps in a log for a single represents an entire solution for the problem. Hence, for our purpose, it is sufficient to have SimStudent perform only those solution steps: if SimStudent cannot correctly perform a step, it is a prediction that the real student would also fail to perform that step correctly on the first attempt.

4.3. Result: Analysis of prediction

59 students solved at least 20 problems. On average, there were a total of 116.8 correct and buggy steps in the 15 training problems, and 55.6 correct steps in the five test problems.

There were 10 different skills in the log data. One skill appeared in the training problems very rarely (27 times total across the 59 students) and hence was excluded from the analysis. The remaining nine skills included five skills for the action steps (add, subtract, multiply, divide, and clt, which is to "combine like terms") and four skills for the type-in steps (add-typein, subtract-typein, multiply-typein, and divide-typein).

We first show the overall performance of problem-solving attempts and then discuss the analysis of predicting real students' performance.

4.3.1. Learning curve – how well did SimStudent learn cognitive skills?

Figure 2 shows the average ratio of successful attempts at performing steps in the test problems, aggregated across all skills and students. The x-axis shows the number of opportunities that SimStudent had had to learn the particular skills whose performance is shown on the y axis at that point.

The average ratio of successful attempts increased as the opportunities of learning increased. After training for 16 times, more than 80% of the steps in the test problems were performed correctly.

This result shows that SimStudent did actually learn cognitive skills from the tutor-interaction log of the real students. Below, we will explore two further issues: How many correct steps in the test problems were predicted as correct? Did SimStudent learn overly general rules that had a tendency to perform steps incorrectly?

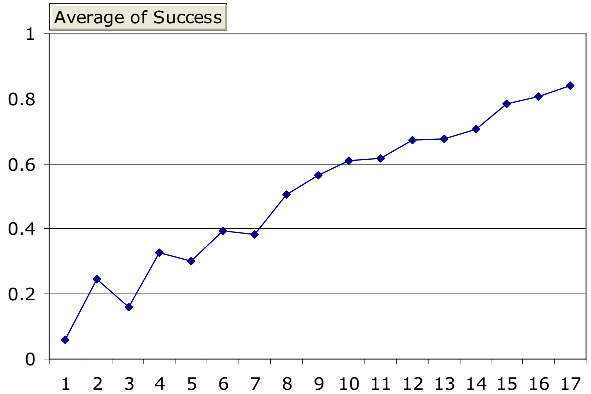


Figure 2: Learning curve showing average ratio of successful problem-solving attempts aggregated across all skills and all students. The x-axis shows the number of opportunities for learning a skill

4.3.2. Analysis of errors in predicting real students performance

The primary purpose of this study is to *predict* real students' performance. Hence SimStudent should not only perform correctly on the steps in which the real students performed correctly but also fail to perform steps in which the real students failed. Thus,

a better analysis is to count the number of matches between the real students' and SimStudent's performance on the test problems.

We define the result of a prediction as follows: first, the result of performing a step is a *success* if there is a production rule fired that reproduces the step correctly; and an *error* otherwise. The real student's step is a *success* if he/she performed the step correctly at the first attempt; and an *error* otherwise[3]. We then define a prediction to be (1) *true positive* (TP), if both SimStudent and the human student's performance were a success, (2) *false positive* (FP), if the SimStudent's attempt was a success while the student's performance was an error, (3) *false negative* (FN), if the result of the SimStudent's attempt was an error, but the student's performance was a success, and (4) *true negative* (TN), if both the result of the SimStudent's and the student's performance were an error. We define the following measures: *Accuracy* = (TP+TN)/(TP+FN+FP+TN), *Precision* = TP/(TP+FP), *Recall* = TP/(TP+FN), *E[rror]-Precision* = TN/(TN+FN), *E[rror]-Recall* = TN/(TN+FP). In this study, we are particularly interested in E-Precision and E-Recall, because SimStudent ought to model not only the real student's successful performance, but also errors.

Figure 3 shows the result of the analysis of predicting real students' performance. As in Figure 2, the x-axis again shows the number of opportunities to learn a skill. The y-axis shows the average for each measure aggregated across all students and skills. As conjectured from the result of the previous section, the Recall increased as SimStudent was trained on more steps – showing the ability for SimStudent to learn rules that perform steps in which the real students succeeded.

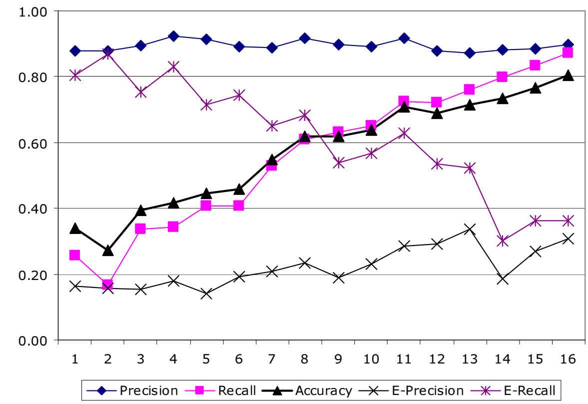


Figure 3: An analysis of predicting real students' performance. The X-axis shows the frequency of learning. The Y-axis shows the average of each measurement aggregated across all students and all skills.

That the Precision stayed high regardless of the number of training examples is rather surprising. It turned out, however, that the real students in the current particular log data made only a very few error steps – only 15% of the steps in the test problems were error steps and the ratio of error steps was stable across the different frequencies[4].

Finally, and most importantly, the E[rror]-precision increased slightly, but overall it stayed low. This means that SimStudent did not accurately predict real students' error steps. To our surprise, the E[rror]-recall decreased as learning progressed. This indicates that as learning progressed, SimStudent tended to learn more production rules that correctly performed those steps that were performed incorrectly by human students. In

[3] This includes cases where the student requested a "hint" on the step.

[4] This does not mean that the real students did not learn. Indeed, the *number* of error steps decreased, but the ratio of error steps to the correct steps stayed the same – there was always about a 15% chance that the real students made an error step in this particular data set.

other words, the current implementation of SimStudent is not correctly predicting human students' erroneous performance.

4.3.3. Analysis of errors of commission

One of the key concerns in the previous analysis is whether and how often SimStudent learned overly general rules. To address this issue, we asked SimStudent to show next steps that can be achieved for each of the steps in the test problems, and assess the correctness for each of these steps. Such evaluation is quite costly, because it requires an oracle for the judgment. We have not implemented the oracle for the analysis just yet. Instead, we have manually gauged the correctness of the rule firing by inspecting the conflict set of the production rules each time a step is about to be performed by SimStudent. We have randomly selected three real students from the log data for this analysis. There were a total of 102 steps in 15 test problems where 6 skills (add, divide, subtract, add-typein, divide-typein, and subtract-typein) were tested.

	Rule firing	
	True Firing	False Firing
add	146	22
add-typein	15	47
subtract	145	16
subtract-typein	13	15
divide	19	35
divide-typein	14	9
Total	352	144

Table 1: Total number of rules in the conflict set for each of the skills. The *False Firing* shows the number of incorrect rule applications – i.e., if applied, the production rule generated an incorrect step.

Table 1 shows the total number of rules appearing in the conflict set during the test. *True Firing* shows the number of production rules that, if applied, generate a correct step. *False firing* shows the number of production rules that, if applied, generate an incorrect step. On average, there were one or two overly general rules, for each of the steps in the test problem, that lead to a wrong step. More surprisingly there were very many opportunities for "add" and "subtract" rules to be applied correctly. This observation agrees with the high precision mentioned above. It also explains why the E[rror]-recall rapidly decreased as the learning progressed – SimStudent quickly learned those rules, which resulted in covering more steps that the real students failed to perform correctly.

5. Conclusion

Using a genuine tutor-interaction log as "demonstrations" of real students performing their skills, SimStudent did learn to generate a student model that can predict more than 80% of the *correct* steps performed by real students.

However, it turned out that SimStudent does not accurately predict real students' *incorrect* steps. Predictions are produced by a student model that is overly general hence, by definition, covered more steps than it ought to cover – SimStudent correctly performed steps that were not performed correctly by real students (low E[rror]-precision). Also, SimStudent learned rules that correctly perform steps, but in a different way than the ones performed by the real students.

Can we use SimStudent as a pedagogical component for an intelligent tutoring system? Currently, the answer leans toward the negative. We are still exploring the issues. One problem may be an incorrect model of students' prior skills. The solution may require different learning methods; the current SimStudent is designed for fast construction of cognitive models using programming by demonstration, not for student modeling.

References

[1] Matsuda, N., W.W. Cohen, and K.R. Koedinger, *Applying Programming by Demonstration in an Intelligent Authoring Tool for Cognitive Tutors*, in *AAAI Workshop on Human Comprehensible Machine Learning (Technical Report WS-05-04)*. 2005, AAAI association: Menlo Park, CA. p. 1-8.

[2] Koedinger, K.R., V.A.W.M.M. Aleven, and N. Heffernan, *Toward a Rapid Development Environment for Cognitive Tutors*, in *Proceedings of the International Conference on Artificial Intelligence in Education*, U. Hoppe, F. Verdejo, and J. Kay, Editors. 2003, IOS Press: Amsterdam. p. 455-457.

[3] Cypher, A., ed. *Watch what I do: Programming by Demonstration*. 1993, MIT Press: Cambridge, MA.

[4] Matsuda, N., et al., *Evaluating a Simulated Student using Real Students Data for Training and Testing*, in *Proceedings of the International Conference on User Modeling*. 2007; in press.

[5] Ikeda, M. and R. Mizoguchi, *FITS: a framework for ITS - a computational model of tutoring.* International Journal of Artificial Intelligence in Education, 1994. **5**: p. 319-348.

[6] Baffes, P. and R. Mooney, *Refinement-Based Student Modeling and Automated Bug Library Construction.* Journal of Artificial Intelligence in Education, 1996. **7**(1): p. 75-116.

[7] Langley, P. and S. Ohlsson, *Automated cognitive modeling*, in *Proceedings of the Fourth National Conference on Artificial Intelligence*. 1984, AAAI: Melon Park, CA. p. 193-197.

[8] MacLaren, B. and K.R. Koedinger, *When and why does mastery learning work: Instructional experiments with ACT-R "SimStudents".* in *Proceedings of the 6th International Conference on Intelligent Tutoring Systems*, S.A. Cerri, G. Gouarderes, and F. Paraguacu, Editors. 2002, Springer-Verlag: Berlin. p. 355-366.

[9] Anderson, J.R., *Rules of the mind*. 1993, Hillsdale, NJ: Erlbaum.

[10] Muggleton, S. and C. Feng, *Efficient Induction Of Logic Programs*, in *Inductive Logic Programming*, S. Muggleton, Editor. 1992, Academic Press: London, UK. p. 281-298.

[11] Koedinger, K.R. and A. Corbett, *Cognitive Tutors: Technology Bringing Learning Sciences to the Classroom*, in *The Cambridge Handbook of the Learning Sciences*, R.K. Sawyer, Editor. 2006, Cambridge University Press: New York, NY. p. 61-78.

[12] Friedman-Hill, E., *Jess in Action: Java Rule-based Systems*. 2003, Greenwich, CT: Manning.

[13] Quinlan, J.R., *Learning Logical Definitions from Relations.* Machine Learning, 1990. **5**(3): p. 239-266.

Data Mining

This page intentionally left blank

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Investigating Generative Factors of Score Matrices

Titus Winters [a,*], Christian R. Shelton [a] and Tom Payne [a]
[a] *UC Riverside, Riverside CA, USA*

Abstract. An implicit assumption in psychometrics and educational statistics is that the generative model for student scores on test questions is governed by the topics of those questions and each student's aptitude in those topics. That is, a function to generate the matrix of scores for m students on n questions should rely on each student's ability in a set of t topics, and the relevance of each question to those topics. In this paper, we use educational data mining techniques to analyze score matrices from university-level computer science courses, and demonstrate that no such structure can be extracted from this data.

Keywords. Cognitive Science, Clustering, Dimensionality reduction, Educational Data Mining

1. Introduction

Over the past century, psychometrics has utilized mathematical techniques to extract meaning from collections of educational data. One of the most common data types in psychometrics is the score matrix, storing the scores for a collection of students on each question from a test. Commonly, educational researchers will apply algorithms developed in the early 20th century to discover the structure or generative model of these matrices. While advanced for their time, and sufficient for some problem types, the great advances made in machine learning may have surpassed these techniques, and may even invalidate some of the assumptions made by early psychometric techniques.

This paper presents an investigation into the generative model of score matrices recorded in computer science (CS) courses in our department. Here we apply machine learning techniques to this canonical psychometric problem and show that on some naturally-arising datasets, the assumptions made for the past century may not be valid.

Specifically, this paper investigates the assumption that a generative model for a score matrix should take into account the topics being covered on the test. From our investigations on course data, the factors that are considered most important by machine learning algorithms consistently have no relationship with human-identified topics.

2. Psychometrics

At its root, the quantitative branch of education that is most applicable to machine learning and educational data mining is *psychometrics*. Psychometrics focuses on measuring

*Correspondence to: Tel.: +1 909 262 0385; E-mail: titus@cs.ucr.edu.

Table 1. Descriptions of datasets. MC indicates multiple choice questions, FR indicates free-response questions subjectively graded, PC indicates partial-credit.

Data set	Course	m	n	Question Types	Scores
008	Intro Computing	269	80	MC	Binary
010	CS 1	190	68	MC	Binary
141	Algorithms	19	15	FR	PC
153	O. S.	66	34	MC,FR	Binary,PC
164	Networks	23	72	MC,FR	Binary,PC
Academic	Online	267	40	FR	Binary
Trivia	Online	467	40	FR	Binary

latent quantities of knowledge and ability in the human mind [1]. These are difficult values to quantify, as there is no direct way to measure them and no implicit units or dimensionality to such a measurement.

One of the most important developments in psychometrics was the development of common factor analysis (CFA), which became the primary area of research in psychometrics for half a century [2]. The dominant theory at the time was that intelligence was univariate (g-theory). However, some data simply did not match the proposed model. Utilizing then-new techniques for calculating correlations between measured values, Charles Spearman developed common factor analysis, which is still used today to understand large quantities of data by discovering the latent factors giving rise to that data. CFA assumes that a linear model of f factors can describe the underlying behavior of the system, in this case the ability of students to answer items correctly: $S_{i,j} = \mu_j + \sum_{k=1}^{f} W_{i,k}H_{k,j} + \epsilon$, where $S_{i,j}$ is the score for student i on item j, μ_i is a hidden variable related to the difficulty of the item, W is a matrix giving the ability of each student with respect to each factor, H is a matrix relating each factor to each item, and ϵ is the noise in score measurement. Generally, the factors in CFA are assumed to be the topics of the various questions.

This paper is an attempt to verify the CFA assumption that the underlying factors in the generative model for a score matrix are the topics of the questions being tested.

3. Datasets

Each of the datasets evaluated in this paper has two components. Most important is the score matrix S, an $m \times n$ matrix of the scores for m students on n questions. We also have a collection of human-generated sets identifying which of the questions (columns of S) can be viewed as testing the same topic. We currently have two sources for datasets: real course data recorded at the per-question level, and an online quiz system. These datasets are summarized in Table 1.

We are primarily concerned with datasets from real courses. Over the past several terms we have recorded detailed score information for a number of courses across our curriculum, ranging from large non-major service courses to upper division electives.

For each of these courses, we have asked the instructor of the course, along with any other regular teachers of the course, to provide us with a set of question groups they would consider grouped by topic. To do this they are provided the text of the questions, but not the scores. They were allowed to place each question in as many groups as they choose. This could be none if they felt the question is not connected to any other questions in the course, or more than one group if a question relates to more than one topic.

The analysis in this paper relies on the presence of this topic information provided by instructors. However, as it is unknown whether the score matrices for actual courses really contain topic information, we felt it prudent to develop datasets with known dependence on topic. To this end we built two additional datasets, one derived from trivia questions from the popular game Trivial Pursuit [3], and one derived from questions from study guides for SAT Subject Tests [4]. Both quizzes are forty questions drawn from four topics, with ten questions in each topic.

The trivia quiz draws questions from Sports and Leisure, Science and Nature, Arts and Entertainment, and Literature. These topics were chosen from the six categories in Trivial Pursuit because they were felt to be the most independent from one another. The questions were drawn from the 20th Anniversary edition of Trivial Pursuit, which focuses primarily on the 1980s and 1990s. As our target audience for the quiz was college students, we felt this was more likely to result in correct answers than questions from other editions. Additionally, we chose only questions that had short one or two word answers, as the online quiz system we developed is a free response system and we wanted to minimize issues stemming from poor spelling.

The academic quiz draws questions from published study guides for the SAT Subject Tests. The SAT Subject Tests are taken by students in their final year of high school, and are used to demonstrate the depth of knowledge that a student has attained in particular areas. There are over twenty test subjects available. Again we chose subjects that we felt were maximally independent: Math, French, Biology, and World History.

Our assumption was that the trivia quiz could function as a baseline for topic extraction on data with little or no actual correlation within each topic. Answering any of these questions correctly is a matter of isolated fact retrieval, and not a test of any deeper knowledge or skill. As such, even though there is an obvious "correct" answer when grouping these questions, we did not expect to be able to extract that answer with any of our candidate algorithms.

The academic quiz represents the opposite end of the spectrum: the questions that were included come from subjects that are generally studied for at least one academic year in high school, and possibly several more in college. The questions require a fair depth of knowledge, and are more topically connected than the trivia questions. Our expectation was that any algorithm that can succeed in topic clustering will have at least partial success on this dataset.

The quizzes were administered online over the course of about a week. The trivia quiz was completed by 467 different visitors, the academic quiz was completed by 267. Correct answers were immediately reported to the test taker, incorrect answers were reported only as "probably" incorrect. The full text input for any incorrect answers was recorded, allowing for post-testing processing and correction for unexpected formatting or spelling.

4. Unsupervised Clustering

Although we performed several experiments on this data, due to space constraints we only present results for unsupervised clustering. Interested readers are invited to seek additional details on this work in [5]. We have named the underlying problem here *topic clustering*: given S and the human-generated groups, is there an unsupervised machine learning algorithm that can generate clusters of the questions in S similar to the human-

generated answer? As discussed in [5], a solution to the topic clustering problem that works on course datasets like the ones discussed here can be used in a wide range of assessment scenarios.

4.1. Experiment

To evaluate the output of our candidate algorithms in topic clustering, we focus on question pairings. Each algorithm produces a set of groups of questions that are being claimed to belong to the same topic. Each dataset has a similar set of groups that are correct or acceptable answers. For the course datasets these were produced by the course instructors, for the quiz datasets these are the topics from which the questions were drawn.

To measure the similarity of the correct answer and the algorithm's results, we create the list of all pairs of questions that were placed in the same group by the algorithm[1], and the list of all pairs of questions that were grouped together by a human. Given these lists we can evaluate *precision* and *recall* for each algorithm. Precision is the percentage of the pairs reported by the algorithm that are present in the correct answer. Recall is the percentage of pairs in the correct answer that were also reported by the algorithm. Thus precision measures how accurate the answer is, and recall measures how complete the answer is. These are independent: grouping only two questions, and grouping those correctly, produces perfect precision but very low recall. Grouping all questions together into one group provides low precision (equal to random chance), but perfect recall.

Further, all but one of the algorithms can produce a numeric certainty on each question's group membership. By varying the cutoff threshold on that certainty, we can adjust the size of the groups reported by each algorithm. If an algorithm's underlying model is accurate, then at least the most-certain questions should be correctly grouped. By evaluating precision and recall across the range of the certainty, we can generate precision-recall curves for each algorithm, demonstrating the accuracy of the algorithm across the certainty range.

4.2. Algorithms Surveyed

The algorithms considered in this study fall into three main groups: clustering algorithms, dimensionality reduction algorithms, and algorithms from educational statistics. Here we list the algorithms, how they are used to produce the certainty groups underlying the precision-recall curves, and describe those that readers may be unfamiliar with.

4.2.1. Clustering

We are evaluating k-means [6], spectral clustering [7] single-linkage [8], complete-linkage [9], and average-linkage [10] algorithms. All of these can give an ordering on which questions are most certainly grouped by sorting the questions by the distance to their cluster center.

Another set of algorithms that we are investigating is the use of single linkage, average linkage, or complete linkage agglomerative clustering on the *correlation matrix* of the questions, rather than working with the raw data itself. In this form, the pair of questions or question clusters to be merged at each step is given by the maximal entry in

[1] Stochastic algorithms were run to convergence and restarted a minimum of 500 times per dataset.

the correlation matrix. The three agglomeration techniques differ in how they merge the rows and columns for the chosen clusters for the next step. Due to space considerations, these are detailed in [5].

4.2.2. Dimensionality Reduction

Dimensionality reduction algorithms can also be used to find the underlying structure of the data. For dimensionality reduction, we evaluated Singular Value Decomposition (SVD) [11], Independent Component Analysis (ICA) [12], and a non-negative matrix factorization (NNMF) [13].

We also evaluated slight alterations of SVD and ICA. In the base versions of these algorithms group membership is determined by the maximum absolute value in the output vector corresponding to each question. In the modified versions we use k-means clustering on the reduced matrices, setting k to t (the number of factors) and assigning group membership for each question according to the resulting clustering.

4.2.3. Educational Statistics

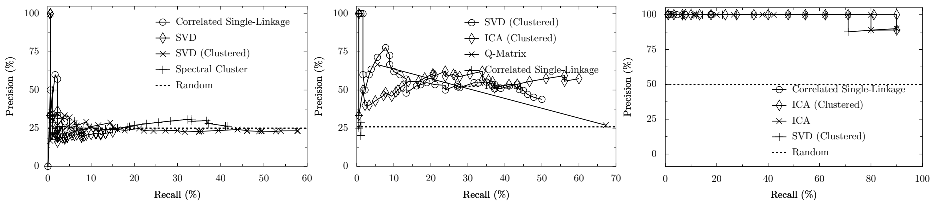
We are also investigating two methods from educational statistics in this study: Common Factor Analysis [2], which we are using as the major point of comparison for educational statistics, and Q-Matrix [14]. Both of these have been specifically suggested by researchers in education for this problem. Q-Matrix is another method of capturing the underlying factors and abilities that give rise to a matrix of student scores. In the most simple case, Q-Matrix is applied to a binary $m \times n$ score matrix.

Q-Matrix operates on this matrix along with the assumed number of underlying abilities or factors, t. Each question is assumed to have a binary relationship with each of the t factors. Each student is similarly assumed to have a binary ability in those factors. Thus Q-Matrix is decomposing the input matrix into a binary $m \times t$ matrix of student abilities in each of those factors, as well as a binary $t \times n$ matrix of question relevance. A student is assumed to answer a question correctly if and only if the factors relevant to that question are a subset of their own abilities.

Most or all Q-Matrix solvers utilize gradient descent on the reconstruction error of the matrix in order to find the optimal model to fit the data. Given an initial random configuration, the algorithm finds the single best entry to toggle in the student matrix or the question matrix in order to best reproduce the input matrix. As a discrete optimization problem with a search space growing as $O(2^{t(m+n)})$, this is computationally intensive.

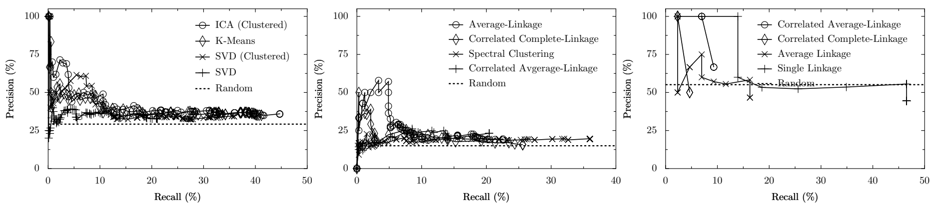
4.3. Results

First we need to confirm our assumptions about the ability for topic information to be extracted from the baseline data. We would like to confirm that the algorithms we are surveying do *not* extract clusters related by topic on the trivia data. This is confirmed in Figure 1a: even the four best algorithms applied to this dataset fail to significantly outperform random chance. Similarly, we would like to see that the questions in the academic dataset do cluster by topic. This is partially confirmed, as seen in Figure 1b. However, even the highest-performing algorithms appear to only be performing at around 50% precision. Further investigation demonstrates that the Math and French portions of the academic dataset can be correctly clustered by almost all of our surveyed algorithms,



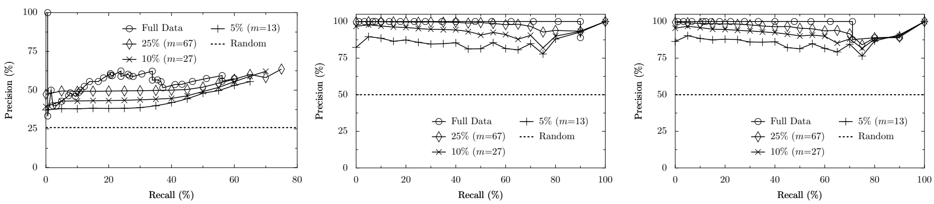
(a) Trivia (b) SAT (c) SAT: Math & French

Figure 1. Precision-recall curves for best four algorithms on baseline datasets



(a) CS 008 (b) CS 010 (c) CS 141

Figure 2. Precision-recall curves for best four algorithms on three course datasets, four factors



(a) SAT (b) SAT: Math & French (ICA)(c) SAT: Math & French (SVD)

Figure 3. Precision-recall curves on for best algorithm(s) on baseline data, varying population sizes

as shown in Figure 1c. The Biology and World History questions behave like the trivia data. Since SAT-level questions on Biology and World History are generally retrieval of isolated facts, more than testing of deeply learned skills, this is less surprising in retrospect.

On data where topic is clearly a factor (the Math and French) we have shown that almost any machine learning technique will acceptably separate the data. For data where topic is not expected to be a factor (the trivia data), all of the algorithms surveyed perform only as well as random chance. One important question is, “Where on this spectrum do course data fall?” Figure 2 is an example of this on three of our CS course datasets with $t = 4$: although there is possibly some dependence on topic, in general even the *best* algorithms are performing no better than chance. Evaluating across the range of t and on all five course datasets shows graph after graph of similar behavior. Given this evidence, topic has little to do with scores on the course datasets.

An important difference between the course data and the baseline data is the size of the datasets: the baseline data has significantly larger student populations than the course data. It is possible that topic clusters can be more accurately extracted if student

Table 2. Average Precision within given Recall Ranges on Course Datasets, 8 factors

Algorithm	0 to 20	20 to 40	40 to 60	60 to 80	80 to 100	AVG
NNMF	**33.9**	**41.2**				**33.9**
Complete-Link (corr)	33.8	17.5				33.1
Random	30.2	30.2	30.2	30.2	**30.2**	30.2
Complete-Link	28.8	36.1	**37.4**			29.8
ICA (cluster)	28.0	38.0				29.5
SVD (cluster)	27.0	37.7				28.9
ICA	28.4					28.4
Average-Link (corr)	28.3					28.3
Spectral	28.1	23.4		26.0	27.2	27.6
k-means	26.4	30.0				27.3
Average-Link	26.8					26.8
CFA	27.6	20.8				25.9
Single-Link	24.4	26.4	22.6	23.1	22.0	24.4
Single-Link (corr)	24.3					24.3
SVD	23.4					23.4
Q-Matrix	19.1	21.4		**31.8**		22.4

populations were larger. Since it is not possible to increase the student population in a real course, instead we test the converse of this hypothesis: "Can we still extract topic information from reduced student populations on the academic dataset?" This is shown in Figure 3. For student populations as small as about 20 students, topic information can be extracted well from the Math and French data. This strongly suggests that it is not the size of the input that causes the poor performance in Figure 2.

Summarizing all of the algorithms on all the course datasets, and setting $t = 8$ (the largest workable value for some of the smaller datasets), we get Table 2. All of the precision values falling within the given range for recall across all course datasets are averaged together and provided here. A blank entry indicates no evaluation of the output of that algorithm had a recall in that range. By setting t as large as possible, we minimize issues of forcing multiple topics into the same group, at the expense of lower recall values. Here we see that nothing significantly outperforms random chance, with the possible exception of the Non-Negative Matrix Factorization. On the course data, it does not appear that topic is a prime factor in the generative model for student scores.

5. Conclusions

Neither experiment presented here successfully shows the dependence of scores on topic information for course data. This contradicts some of the fundamental assumptions in psychometrics. We have provided strong evidence that on common types of data (namely, scores collected in university courses), the generative model for those scores is, in fact, *not* based on the topics for those questions. In our informal evaluation of the extracted groupings we have found anecdotal evidence of the importance of time (which questions were asked on the same day), trick questions (which intro CS questions the authors have difficulty answering), and many groups whose relation is completely unclear. This means that for individual questions in such a course, it appears to be more important that the student is having a good day, or is good at trick questions, or is generally good in the course, rather than whether they know a lot about the topic of a particular question.

The implications of this can be interpreted in a variety of ways, and the verification or validation of those implications is far beyond the scope of this paper. Our anecdotal take-away belief from this experiment is that if in-course assessment is intended to measure accurately a student's knowledge in each topic covered in a course, it appears that many courses are performing insufficient assessment. This could be insufficient due to small number of questions in each topic, or insufficient attention paid to construction of assessment instruments.

We feel that topic clustering is an important general problem in the field of educational data mining. A significant percentage of EDM papers are involved in something like topic clustering, often on ITS data or scores augmented with metadata like timing. The converse view of topic clustering is to cluster students based on similar proficiencies in the reduced $m \times t$ matrix implied by CFA or similar models. Student clustering is perfectly suited to provide summary aggregation in post-analysis reports. For these reasons and the assessment reasons presented in [5], we feel topic clustering is a critical topic for EDM, and one that has significant implications on educational statistics and pedagogy.

The datasets used in these experiments are publicly available on the lead author's website. We encourage other researchers to explore their own techniques for extracting information from such course matrices.

References

[1] Karl Pearson. *The life, letters and labours of Francis Galton*. University Press, 1914.

[2] Charles Spearman. General intelligence, objectively determined and measured. *American Journal of Psychology*, 15:201–293, 1904.

[3] Trivial pursuit. http://www.trivialpursuit.com, 2006.

[4] SAT subject tests. http://www.collegeboard.com/student/testing/sat/about/SATII.html, 2006.

[5] Titus Winters. *Educational Data Mining: Collection and Analysis of Score Matrices for Outcomes-Based Assessment*. PhD thesis, UC Riverside, May 2006.

[6] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of 5th Berkely Symposium on Mathematical Statistics and Probability*, pages 281–297, 1967.

[7] I. Fischer and J. Poland. Amplifying the block matrix structure for spectral clustering. In M. van Otterlo, M. Poel, and A. Nijholt, editors, *Proceedings of the 14th Annual Machine Conference of Belgium and the Netherlands*, pages 21–28, 2005.

[8] R. Sibson. Slink: An optimally efficient algorithm for the single link cluster methods. *The Computer Journal*, 16(1):30–34, 1973.

[9] D. Defays. An efficient algorithm for a complete link method. *The Computer Journal*, 20(4):364–366, 1977.

[10] R. R. Sokal and C. D. Michener. Statistical method for evaluating systematic relationships. *University of Kansas science bulletin*, 38:1409–1438, 1958.

[11] J. C. Nash. The singular-value decomposition and its use to solve least-squares problems. In Adam Hilger, editor, *Compact Numerical Methods for Computers: Linear Algebra and Function Minimisation, 2nd ed*, pages 30–48, 1990.

[12] Aapo Hyvärinen. Survey on independent component analysis. *Neural Computing Surveys*, 2:94–128, 1999.

[13] Daniel D. Lee and H. Sebastian Seung. Algorithms for non-negative matrix factorization. In *NIPS*, pages 556–562, 2001.

[14] M. Birenbaum, A. Kelly, and C. Tatsuoka. Diagnosing knowledge state in algebra using the rule-space model. *Journal for Research in Mathematics Education*, 24(5):442–459, 1993.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

How did the e-learning session go? The Student Inspector[1]

Oliver SCHEUER and Claus ZINN [2]
German Research Center for Artificial Intelligence (DFKI)

Abstract. Good teachers know their students, and exploit this knowledge to adapt or optimise their instruction. Traditional teachers know their students because they interact with them face-to-face in classroom or one-to-one tutoring sessions. In these settings, they can build student models, *i.e.*, by exploiting the multi-faceted nature of human-human communication. In distance-learning contexts, teacher and student have to cope with the lack of such direct interaction, and this must have detrimental effects for both teacher and student. In a past study we have analysed teacher requirements for tracking student actions in computer-mediated settings. Given the results of this study, we have devised and implemented a tool that allows teachers to keep track of their learners' interaction in e-learning systems. We present the tool's functionality and user interfaces, and an evaluation of its usability.

Keywords. Methods, tools and techniques for effective evaluation of cognitive, meta-cognitive and affective issues, distance-learning contexts, log data analysis.

1. Motivation

There is an increasing use of technology in education. Industrial-strength e-learning systems provide teaching material that can be accessed anywhere and anytime, especially outside classroom sessions. While e-learning systems can offer many advantages – the presentation of high-quality multimedia content, the self-paced exploration of learning material, the engagement of learners in interactive exercises, the opportunity to communicate and collaborate with other learners *etc.* – there are also some dangers. Learners may not be able to use the new technology effectively, potentially by spending more time and mental resources in struggling with the technology than with the actual learning process. Also, parts of the technology could be abused, *i.e.*, the misuse of the communication and collaboration tools for extensive off-topic chit-chat. The absence of teacher authority may in particular harm learners with limited abilities to self-direct their learning.

Only teachers that are aware of these and other dangers can intervene – if only there were a STUDENT INSPECTOR that would allow teachers to getting to know their learners in distance learning contexts. A tool that could capture learners' individual learning paths (and dead-ends), could keep track of learners' knowledge, higher-level competences, or the lack thereof, and that could list their problems with the subject domain taught.

---

[1]The work presented in this paper is supported by the European Community under the Information Society Technologies programme of the 6th Framework Programme for RTD – project iClass contract IST-507922.

[2]Correspondence to: Oliver Scheuer, German Research Center for Artificial Intelligence (DFKI), 66123 Saarbrücken, Germany, Tel.: +49 681 302 5377; Fax: +49 681 302 2235; E-mail: oliver.scheuer@dfki.de.

We are devising a STUDENT INSPECTOR that analyses log data gathered from e-learning environments and that presents the results of these analyses to teachers. In this paper, we present a prototype implementation that incorporates teachers' needs by combining state-of-the-art log data processing and presentation with AI-based analyses.

2. Background

Some e-learning systems support the tracking, analysis, and reporting of learner-system interactions. Commercial course management systems like WebCT track, *i.e.*, the date of the first and most recent sign-on for each student in the course; the usage of the conference tool (number of articles read, number of postings and follow-ups) to measure involvement and contribution; the use of other tools (*e.g.*, notes, calendar, e-mail); and visits to the content material (page visits, view times) to monitor course usage, usefulness, and quality. Data is then aggregated and visualised in tables or with bar charts [3,7].

The ASSISTment system is a web-based tutoring system in which students work on a set of test items. Correct answers lead to the next item, incorrect ones open a small tutorial session where the original problem is broken down into a series of scaffolding questions aimed at guiding the student. The system offers reports on students' performances. For enabling a fine-grained assessment of students' abilities, problems and scaffolding questions are annotated with the skills they require. The generated reports present, for instance, the time spent in the system, the number of items worked on, the number of items solved, the number of hints requested, strongest/weakest skills of a class or of an individual student, and change of performance over time [2].

DataShop (`www.learnlab.org/technologies/datashop`) is a central repository that offers learning scientists services to store and analyse their data. Generated reports present, *e.g.*, learning curves by student or skill and detailed data for a given problem step (submitted answer, assessment of answer, system response *etc.*).

Merceron and Yacef pursue a data mining approach to student activity tracking [5]. Their tool for advanced data analysis in education (TADA-Ed) offers, for instance, an association rule algorithm that can be used to identify sets of mistakes that often occur together, or a decision tree algorithm to predict final exam marks given students prior usage behaviour. Usage of the tool requires basic knowledge in data mining (and machine learning as its technological basis), and therefore, TADA-Ed is a tool that is primarily directed at educational researchers rather than professional teachers.

Interaction analysis tools are usually seen as an extension to e-learning environments; hence, their reusability is very limited. Moreover, the quality of their analyses crucially depends on the richness of the data logs that e-learning systems produce, which in turn depends on their content representations and the metadata describing them.

Besides the opportunities and limitations of given system characteristics, there are also the desiderata, priorities and preferences of teachers for tracking, analysing, and presenting learner-system interactions. In summer 2006, we conducted a study to better understand teachers' requirements for a tool that helps them understanding their students in distance-learning contexts. Their answers to this questionnaire revealed a clear preference for pedagogically-oriented measures, namely, overall success rate, a learner's mastery levels and typical misconceptions, the percentages of exercises tackled, and amount of material read. Those indicators are quite traditional, and naturally, easy to interpret and work with. Historical information, *i.e.*, past course coverage and activities, learners'

navigational style, and social data attracted less interest, and were judged as too cumbersome to inspect or too time-intensive to interpret (for details, see [8]).

The design of our STUDENT INSPECTOR also profited from the experiences we gained with ActiveMath [4], a learning environment that aims at delivering tools for personalised learning in mathematics. To inform the selection of content, ActiveMath has access to learner modelling and a rich repository of learning objects that are annotated along various dimensions (*e.g.*, the competences they train, the domain prerequisites that they require, their difficulty). Many learner-system interactions are logged, and we have gathered rich data sets that we can analyse along various lines of research.

The true potential for action analysis was realised, however, within the iClass project [6]. iClass aims at empowering teachers with tools to better monitor and manage the progress of their classroom students. iClass also wants to empower learners through self-regulated personalised learning: learners should take more responsibility for their learning, and for this, they need to be supported by tools that inform their decision making (*e.g.*, identification of learning goals, choice of learning scenarios, time scheduling). In this respect, the STUDENT INSPECTOR will be of use for teachers *and* learners.

3. Architecture

Fig. 1 depicts the STUDENT INSPECTOR's architecture. Its organisation into layers minimises dependencies between components and enhances the connectivity of e-learning systems. There are four layers. The **database layer** serves to persistently store interac-

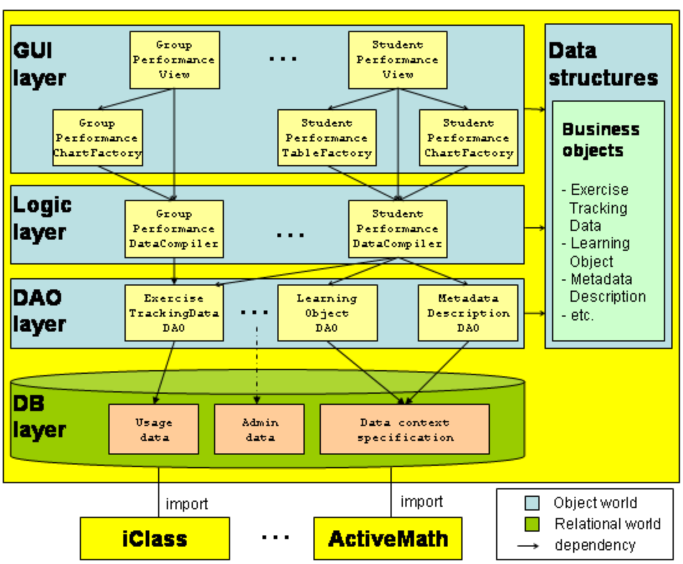


Figure 1. The STUDENT INSPECTOR's architecture.

tion data. Three kinds of data can be distinguished: (i) data describing individual learning objects by means of their metadata profile, the composition of learning objects into course structures, and specifics of the e-learning system (*e.g.*, use of tools); (ii) actual

usage data reflecting learners' activities with the e-learning system (*e.g.*, page views, exercise steps, glossary queries) and system's evaluation of these activities (*e.g.*, exercise success rate, diagnosis of learners' misconceptions); and (iii) administrative data (*e.g.*, learner groups, task assignments). The **data access object layer** converts relational database entries into business objects, and *vice versa*. The Hibernate persistence framework (`www.hibernate.org`) enables this mapping, and also offers other persistence-related services. The **logic layer** defines the logic for the aggregation and compilation of objects, *i.e.*, the analysis of tracking data at the object level, given their intended presentations. The logic layer has also access to machine learning algorithms for more sophisticated analyses. The **GUI layer** consists of views, each of them encapsulating a functionality (*e.g.*, views for overall group performance, student performance along topics). This layer is supported by a program library implementing, *i.e.*, diagrams, charts, and tables.

4. Functionalities

The Student Inspector consists of three major components: a browser to explore data, an admin module to manage students and student groups, and to assign tasks to them, and an AI-based analyser to perform more sophisticated data processing. In this paper, we focus on the modules for browsing and AI-based analyses.

4.1. The Browser

The teachers from our study mostly care about learners' performance, misconceptions, and topic coverage [8]. The Student Inspector implements those and other aspects.

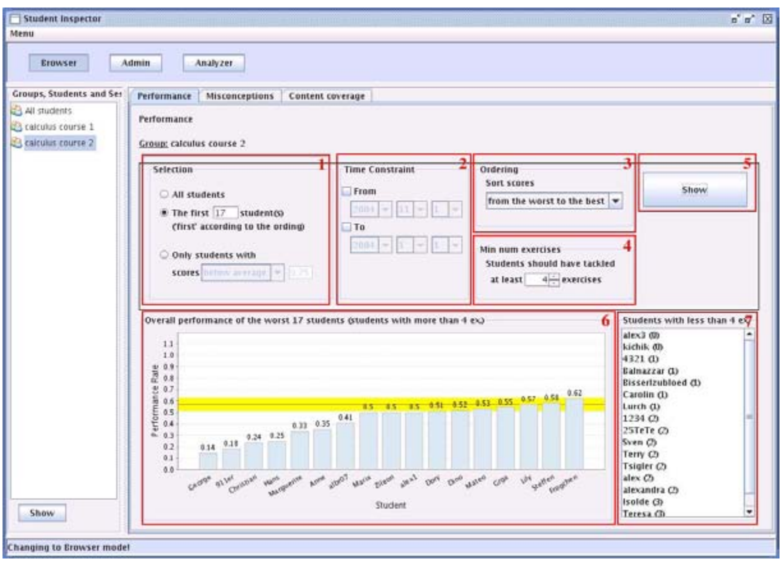


Figure 2. Identifying your classroom's performance.

Performance measurement. Fig. 2 depicts the STUDENT INSPECTOR's presentation of individual and group performance scores. The individual performance of a learner is computed by averaging over her test or exercise scores. Performance values are computed for learners that surpassed the threshold for the minimum number of exercises. Users can use this view to identify the average performance of a group, and consequently those learners that perform below or above the average. Users can focus their performance research on learners (see 1), or on the time of the learning session (see 2). Results can be ordered by student name or performance scores (see 3), and influenced by the exercise threshold (see 4). When these parameters are submitted (see 5), two pieces of information are computed: a chart displaying individual performances and the group's average (see 6), and a list of students not assessed given their insufficient exercise trials (see 7).

Learner Misconceptions. State-of-the-art e-learning systems like ActiveMath can diagnose learner input to exercises to identify typical errors or misconceptions. Fig. 3 displays the STUDENT INSPECTOR's analysis and presentation of such logs for a given learner group. With the pie chart teachers get access to the most frequent errors their learners

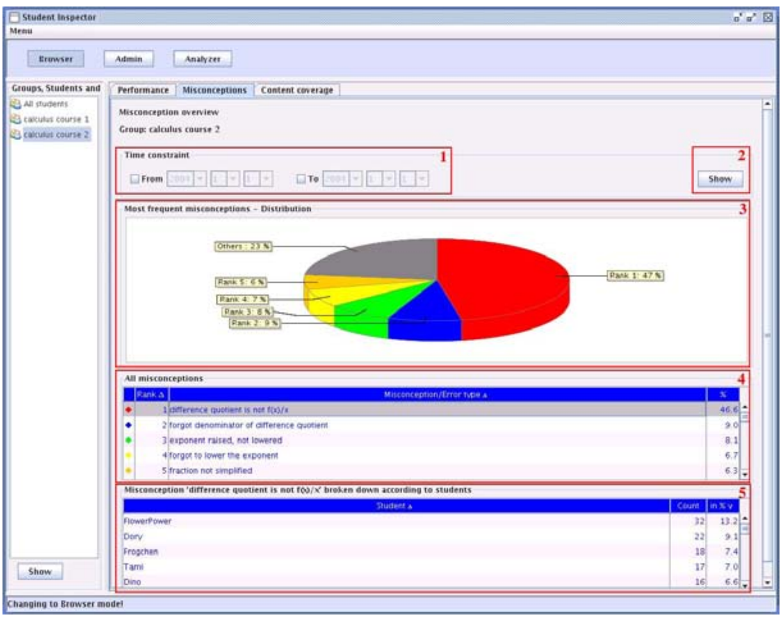


Figure 3. Browsing the misconceptions of your classroom.

showed when working with the e-learning system. In learner group "calculus course 2", *i.e.*, almost half of the learners had difficulties with the difference quotient (see 3). A table (see 4) gives access to a complete and sortable listing of errors as identified by the e-learning system. Moreover, teachers can select a misconception, *i.e.*, a table row, to obtain a list of all learners that made the corresponding error (see 5).

Topic Coverage. E-courses usually cover various topics. A calculus course, *i.e.*, might be about sequences, series, differentiation, and integration. Fig. 4 displays the STUDENT INSPECTOR's analysis and presentation of topic coverage, given a learner's exercise per-

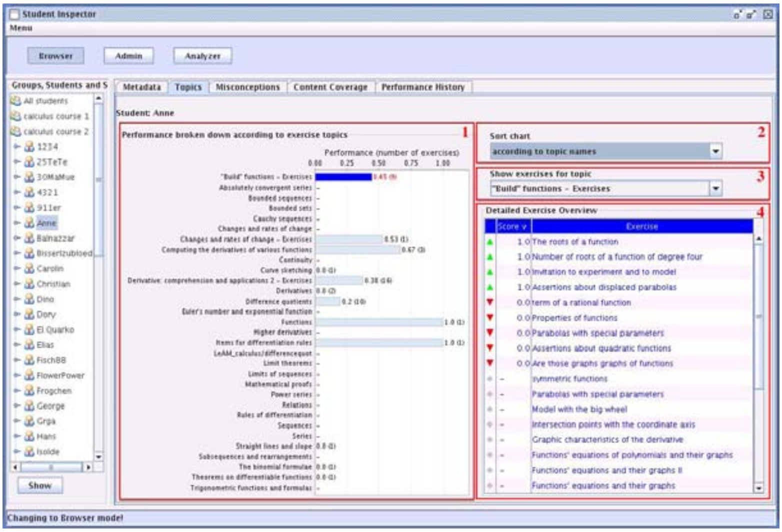


Figure 4. Browsing individual performances via topics.

formance for each of the topics. This view allows teachers to find, *i.e.*, weak and strong topic areas for a given student, here Anne (see 1, 2), and gives also access to the exercises Anne has tackled to achieve her performance score for a selected topic (see 3, 4).

4.2. *The Analyser: Predicting Future Learning Outcomes*

The analyser uses machine learning techniques to predict future learning outcomes. Teachers can ask the exercise recommender to suggest "more difficult" exercises to, say, challenge well-performing learners, or ask for "easier" exercises to, say, motivate below-average learners, given learners' knowledge state and competencies. Analyses can be run for student groups or individuals. To initiate the machine learning method, one first selects those metadata categories judged relevant from the list of available categories. Subsequently, a model is computed given all exercises that the student has tackled. Then, untackled exercises are classified within this model, yielding two tables: the first table shows exercises where a bad performance is expected, whereas the second table shows exercises where a good performance is expected. Expert users can display the model.

5. Evaluation

The STUDENT INSPECTOR has been evaluated. Similar to our first questionnaire, where we gathered teacher requirements, we have devised a second questionnaire to evaluate the implementation of these requirements. Given a series of tool screenshots that demonstrate various analyses and views of student tracking data, we have asked participants to evaluate the software along the lines (adapted from [1]): *usefulness* (does the information support your work); *ease of use* (GUI handling); *functionality* (completeness and sufficiency of information); *organisation of information* (clear or confusing presentation of information); and *terminology* (understandability of wording and labelling). Furthermore, for each of the views we asked whether they would use it in practise.

Questionnaire Participants' Profiles. 31 participants filled in the complete questionnaire, most of them rather experienced with an involvement of 4 or more years in the area of e-learning (72%). The most often mentioned roles were teacher/instructor (79%), instructional designer (57%) and course coordinator (39%). All together, participants used 17 different e-learning systems; the most mentioned ones were Blackboard (11), WebCT (5), Moodle (3) and WTOnline (3). A large majority work in a college or university context (undergraduate courses: 79%, postgraduate courses: 64%). Participants teach a wide range of topics, among others, biology, nursing, statistics, mathematics, engineering, psychology, educational science and information technology.

Questionnaire Results. Fig. 5 depicts the average assessment for the views presented in this paper as well as for the participants' overall judgment of the tool. All four views were well received, with average scores near or above 4 (out of a maximum of 5 points). Also, the tool as a whole got good marks. The positive feedback was strengthened by the fact that the large majority of participants would use each view in practise.

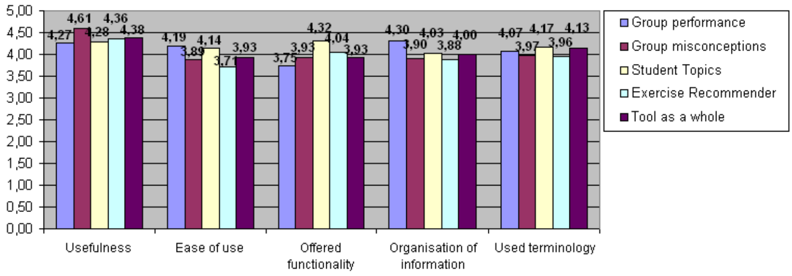


Figure 5. Average scores for selected views.

Information about a group's misconception received the best average usefulness score among all views. One teacher found the pie chart visualisation of misconceptions "quit [sic] helpful, especially with a large class", and that it allows one "to find out the most difficult problems easily". A participant noted that he could "evaluate [his] type of work better with this [misconception] view". This information was found to be "potentially very good for modifying instructions for exercises, or providing additional material". Another participant wished they had such a tool, "especially for our professors who are not very sensitive to students' needs".

The group performance view was also well received. Participants "liked especially the graphic presentation of the data", or even "like[d] this interface better than BlackBoard"; we received similar remarks also to other views. Suggested improvements for this part included a function to "email individuals highlighted", or to "drill down from here - to look at an individual students' results for all assessments to find out why their mark is high or low". The topic-specific performance view was rated of "great value!". It would "develop some useful information fore [sic] changing the course".

Participants were concerned about the AI-based analyser: it is "too complicated for a teacher to handle", presuming "heavy up-front preparation on model definition and creation", with unreliable outcomes: "not sure whether I would trust the model to give me correct answers", or whether "it shows me more than the previous, simpler screens".

6. Discussion and Future Work

Overall, the reaction to the STUDENT INSPECTOR was very positive. Teachers welcomed the abilities "to examine student performance trends and specifics", and to inform teachers "to give feedback to the students". Moreover, the visualisaton of information was judged a great benefit, the usability "easy to use". Some teachers feared, however, that the interface could scare those users which are "not in general comfortable with computers". It is clear that the use of all the features is very time-consuming, and that some information is more directed at educational researchers and content developers. Nevertheless, we had participants asking for more functionality, *e.g.*, visualise collaborative processes; keep a record of all skills achieved; use these records to inform group formation; plot learners's progress over several years; and include learners's time consumption per exercise. To satisfy these demands, we may need to customise the STUDENT INSPECTOR for different target groups or preferences, or to add some configuration management. One participant asked for a training component so that teachers could profit from a maximum of available student tracking analyses; others demanded transparency with respect to the computation of performance scores; and some participants were overwhelmed by the large variety of data and metadata, and their potential manual import, or the synchronisation of data between the STUDENT INSPECTOR and their learner management system.

One teacher noted that the STUDENT INSPECTOR is in favour of a class-instruction model, whereas some teachers may be more interested in student-centred, constructivist pedagogy. In fact, we are currently considering learners as a major target group for using our tool. This is in line with the iClass consortium that promotes *self-regulated personalised learning*. Self-directed learners need to inform their choices (*e.g.,* learning goals, styles) by tools that help them to become aware of their learning process, and that can generate personalised learning content to support this process. Teachers will still be necessary, esp. for learners with moderate abilities to self-regulate their learning. Teachers need to know these students to give them direction, and to adapt their teaching to their needs. Thus both parties could profit from the STUDENT INSPECTOR. For this, we are enhancing our tool, continuously checking with end-users that we are on track.

References

[1] J. P. Chin, V. A. Diehl, and K. L. Norman. Development of an instrument measuring user satisfaction of the human-computer interface. In *Proc. of CHI'88*, pages 213–218. ACM, 1988.

[2] M. Feng and N. Heffernan. Toward live informing and automatic analyzing of student learning: reporting in the assistment system. *J. of Interactive Learning Research*, 18(2):207–230, 2007.

[3] M. W. Goldberg. Student participation and progress tracking for web-based courses using webct. In *Proc. of the 2nd North American Web Conference*, Fredericton, NB, Canada, 1996.

[4] E. Melis, G. Goguadze, M. Homik, P. Libbrecht, C. Ullrich, and S. Winterstein. Semantic-aware components and services of activemath. *Brit. Journal of Educational Tech.*, 37(3), 2006.

[5] A. Merceron and K. Yacef. Tada-ed for educational data mining. *Interactive Multimedia Electronic Journal of Computer-Enhanced learning*, 7(1), 2005.

[6] A. Turker, I. Görgün, and O. Conlan. The challenge of content creation to facilitate personalized e-learning experiences. *Int'l Journal on E-Learning*, 5(1):11–17, 2006.

[7] WebCT. *WebCT Vista^{TM} 3.0 Designer and Instructor Reference Manual*, April 2004. Technical Communications, `http://www.webct.com`.

[8] C. Zinn and O. Scheuer. Getting to know your student in distance-learning contexts. In *Proc. of EC-TEL*, volume 4227 of *LNCS*, pages 437–451. Springer, 2006.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Can a Computer Listen for Fluctuations in Reading Comprehension?

Xiaonan ZHANG[1], Jack MOSTOW, and Joseph E. BECK
Project LISTEN, School of Computer Science, Carnegie Mellon University

Abstract. The ability to detect fluctuation in students' comprehension of text would be very useful for many intelligent tutoring systems. The obvious solution of inserting comprehension questions is limited in its application because it interrupts the flow of reading. To investigate whether we can detect comprehension fluctuations simply by observing the reading process itself, we developed a statistical model of 7805 responses by 289 children in grades 1-4 to multiple-choice comprehension questions in Project LISTEN's Reading Tutor, which listens to children read aloud and helps them learn to read. Machine-observable features of students' reading behavior turned out to be statistically significant predictors of their performance on individual questions.

Keywords. Reading comprehension, automated assessment, children's oral reading, cloze questions, speech recognition, Reading Tutor

1. Introduction

Reading has widespread importance in intelligent tutoring systems, both as a means of instruction, and as an important skill to learn in its own right. Thus the ability to automatically detect moment-to-moment fluctuations in a student's reading comprehension would be of immense value in guiding an intelligent tutoring systems to make appropriate instructional decisions, such as estimating student knowledge or determining what points to explain further. This paper explores detection of comprehension fluctuations in Project LISTEN's Reading Tutor [1], which uses speech recognition to listen to children read aloud, and responds with various feedback.

Human tutors track students' comprehension by asking comprehension questions. Similarly, to test students' comprehension, the Reading Tutor occasionally inserts a multiple-choice cloze question for the child to answer before reading a sentence [2]. It generates this question by replacing some word in the sentence with a blank to fill in. Here is an example. The student picks the story *Princess Bertha and the Lead Shoe*. Some time later the Reading Tutor displays the following sentence for the child to read aloud: *"I must go get Herb the Horse before he runs off again," Princess Bertha exclaimed.* The Reading Tutor turns the next sentence into a cloze question, which it displays and reads aloud: *With that, Princess Bertha took off for her ____.* The Reading Tutor then reads the four choices: *gift; friend; lesson; horse*, and waits for the child to click on one of them. It says whether the student chose the right answer,

[1] Corresponding Author: CMU-LTI, 5000 Forbes Avenue, Pittsburgh, PA 15213, USA; E-mail: xiaonanz@cs.cmu.edu.

defined as the original text word, namely *horse*. Then it displays the correctly completed sentence for the student to read aloud. The three distractors are words of approximately the same difficulty as the right answer, selected randomly from the same story. Thus such questions are entirely automatic to generate, administer, and score. Cloze questions test the ability to tell which word fits in a given context. Performance on these automatically generated cloze questions correlates well with scores on a standard test of reading comprehension ability [3].

However, inserting too many comprehension questions wastes time, disrupts the flow of reading, and annoys students. To address this problem, this paper investigates the following question: can a computer listen for fluctuations in comprehending a text? Specifically, can it detect comprehension fluctuations by listening to the student read the text aloud, and tracking help requests in computer-assisted reading?

We answer this question in the context of the Reading Tutor, by using students' observable reading behavior to help predict their performance on individual cloze questions that reflect their fluctuating comprehension. We assume that comprehension, reading behavior, and cloze performance are related as shown in Figure 1.

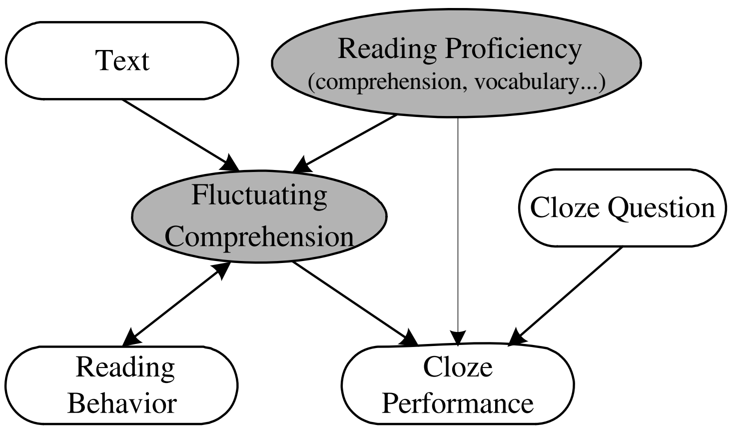


Figure 1: Conceptual model. An arrow from X to Y indicates that X influences Y. Shaded nodes represent hidden entities, while others are observable.

According to this model, comprehension fluctuates as a student with a given level of reading proficiency reads some text whose difficulty for that particular student varies over the course of the text. That is, we model reading proficiency as a relatively stable *trait* that changes only slowly over time, and comprehension as a fluctuating *state*.

The student's fluctuating comprehension is not directly observable, but has observable effects. Comprehension affects (and is affected by) the student's reading behavior, such as speed, prosody, and requests for tutorial assistance. Comprehension also affects the student's performance on cloze questions inserted in the text – but it is not the only influence. Other influences include the student's proficiency as well as the difficulty of the cloze question.

If observable reading behavior really reflects comprehension fluctuations, then it should enable us to predict cloze performance more accurately than if we use only the

proficiency of the student and the difficulty of the cloze question. The central task of this paper is to test this hypothesis.

This work is novel because it focuses on detecting fluctuations in text comprehension, in contrast to previous work on assessing students' general reading comprehension skill [2, 4]. Some work [5] has tracked changes in students' reading skills, but focused on decoding rather than on comprehension, and on growth over time rather than on momentary fluctuations. Other work [6, 7] has used speech-based features to detect fluctuations in cognitive load, but not in reading comprehension. Finally, a substantial body of work [8] has studied transitory comprehension processes, but using eye tracking rather than speech input.

The rest of the paper is organized as follows. Section 2 describes a statistical model of student performance in terms of the conceptual model in Figure 1. Section 3 tests this model and explains the results. Section 4 discusses limitations and potential future directions. Section 5 concludes by summarizing contributions.

2. A Statistical Model of Reading and Cloze Performance

We build on a previous statistical model [9] that focused only on the rightmost three nodes in Figure 1. By accounting for differences in question difficulty, this model was able to infer students' proficiency from their cloze performance more accurately than their raw percentage correct on cloze questions could predict. We extend this model by adding information about students' reading behavior.

The form of our model is Multinomial Logistic Regression, as implemented in SPSS 11.0. Its binary output variable is whether the student answered the cloze question right. We now describe its predictor variables.

Reading proficiency: To capture between-student differences in reading proficiency, we include student identity as a factor in the model, as in [9]. The student identity variable is a unique code assigned by the Reading Tutor when it enrolls a student. This variable also subsumes any other student traits that might affect cloze performance, such as motivation. Including student identity as a factor controls for individual differences between students, and accounts for statistical dependence among responses by the same student instead of treating them as independent [10].

Cloze question difficulty: We adopt the same predictor variables used in [9] to represent features that affect the difficulty of cloze questions. One such feature is the length of the question in words. Another feature is the number of choices with the same part of speech as the right answer. For instance, our example cloze question (*With that, Princess Bertha took off for her__.* [*horse*]) would be easier if all distractors had the wrong part of speech, *e.g.*, *finish*, *come*, and *ordinary*. As a student-specific indicator of difficulty, the model also includes the student's response time to answer the cloze question. Response time reflects student engagement [11] and confidence; hasty response times indicate guesses [2], while long response times reflect uncertainty.

Reading behavior: The new variables in our model characterize students' reading behavior, both their oral reading and the tutorial assistance they receive, when they read the cloze sentence with the right missing word filled in. To derive these variables, we first defined some 30 features that we thought might reflect comprehension fluctuations. These features describe or relate to the words being read, the prosody of the oral reading, the time-aligned output of the speech recognizer, and the assistance given by the Reading Tutor.

The individual features are defined in terms specific to the Reading Tutor and unwieldy to explain. None of them by itself is a dramatically effective indicator of comprehension. They are not independent, and in fact some of them correlate strongly with each other. Moreover, 30 features is enough to pose a risk of overfitting.

To solve this problem, we ran Principal Components Analysis on the 30 features, using the SPSS 11.0 implementation including rotation of components to increase interpretability. We selected the top five components (which together explain about 65% of the variance in the data) because the sixth and subsequent factors account for much less variance. We include these five components (more precisely, standardized component scores of each case on each of these five components) as covariates in our model. As is typical, the principal components do not translate to simple constructs. However, the general nature of each component is easy to describe, and may have analogues in other intelligent tutoring systems. We therefore describe each component in general terms. Space and clarity preclude a comprehensive (and incomprehensible!) list of raw features. Instead, we illustrate each component with one or two underlying features that have loading greater than 0.4 (the component loadings are the correlation coefficients between the features and components). The top five components are:

1. **Sentence coverage:** *e.g.*, the percentage of a text sentence read by the student
2. **Fluency**: *e.g.*, the number of words read per second, and the number of letters in a sequence of words read without pausing longer than 0.5 seconds
3. **Words per utterance**: *e.g.*, the number of text words per utterance accepted by the Reading Tutor as read correctly, and the number of utterances per sentence
4. **Rereading**: *e.g.*, the number of times a sentence is read
5. **Truncation**: *e.g.*, the number of incomplete words output by the recognizer

We also experimented with features related to fundamental frequency (F0), including average, maximum and minimum F0, the slope of the F0 curve, and the pitch range, computed as maximum F0 minus minimum F0. However, they turned out to be insignificant predictors of cloze performance, so we removed them from the feature set.

3. Evaluation

We now evaluate how well the model and various ablated versions of it fit empirical data logged by the Reading Tutor during the 2002-2003 school year. This data includes 7805 cloze responses (72.1% of them correct) by 289 children, ranging from grade 1 to grade 4, at seven public schools in the Pittsburgh area. The logged reading behavior of these students includes the audio files of each student's utterances, the output of the speech recognizer, and the actions of the Reading Tutor, such as help on hard words.

Table 1 summarizes the results of fitting the model to this data set. The table lists the predictor variables in the model, one per row, grouped into the categories discussed in Section 2 above: reading proficiency, cloze question difficulty, and reading behavior. Successive columns of the table show the 8-character variable name used in SPSS; what it represents; a chi-square statistic for the variable (the difference in -2 log-likelihoods between models with and without the variable); the degrees of freedom for the variable; and its statistical significance in the model.

As Table 1 shows, all three groups of variables contain significant predictors, including two of the five composite variables that describe reading behavior. The respective β coefficients of these five variables are .254, .240, .092, -.044, and .008, reflecting their relative strength. The sign represents their qualitative effect on the

chances of answering the cloze question right. Only FAC1 and FAC2 are statistically significant. Their β values reflect positive effects for reading more of the target sentence and reading more fluently.

Table 1. Significance of predictor variables (from SPSS Likelihood Ratio Tests)

Category	Predictor variable	Description of predictor variable	Chi-square	df	Sig.
Reading proficiency	USER_ID	Student identity	655.198	288	.000
Cloze question difficulty	Q_RC_LEN	Length of cloze question in characters	3.650	1	.056
	PERC_Q_P	% of sentence preceding deleted word	8.630	1	.003
	DIFFICUL	Difficulty of target word (4 categories)	64.241	3	.000
	TAG_POS	Target word part of speech	17.268	4	.002
	TPOS_INT	# legal parts of speech of target word	0.009	1	.926
	TAG_PR_M	Target has its usual part of speech	4.718	1	.030
	CONF_POS	# distractors with target part of speech	.140	1	.708
	RTBIN9	Response time (one of 10 bins)	89.571	9	.000
Reading behavior	FAC1	Sentence coverage	43.921	1	.000
	FAC2	Fluency	42.570	1	.000
	FAC3	Words per utterance	2.359	1	.125
	FAC4	Rereading	1.842	1	.175
	FAC5	Truncation	0.073	1	.788

Table 2. Comparison of full and ablated models

Model name	Reading proficiency (student identity)	Cloze question variables	Reading behavior variables	Nagelkerke's R^2	Adjusted R^2	Classification accuracy (%)		
						all	right	wrong
Full model	√	√	√	0.247	0.209	75.6	92.6	31.3
ID+cloze [9]	√	√		0.229	0.190	75.5	93.7	28.2
ID+reading	√		√	0.212	0.175	74.9	93.7	26.3
ID-only	√			0.172	0.134	74.2	95.4	19.3
Cloze-only		√		0.073	0.070	72.7	98.2	6.7
Reading-only			√	0.069	0.068	72.1	97.4	6.3

Table 1 shows the significance of the individual variables. But to understand how the groups of variables contribute to the model, we compare the full model shown in Table 1 to various ablated models that omit one or more of these groups. Table 2 summarizes these models, one model per line, starting with the full model and listed in order of decreasing fit. The first four columns of the table describe which groups of variables the model includes. Nagelkerke's R^2 in logistic regression quantifies the explanatory power of the model, much as R^2 does in linear regression, but with lower typical values. Cross-validation to test generality is problematic due to the student identity variable, so to estimate model fit on unseen data, we modify Nagelkerke's R^2 to take into account the sample size and the number of features in the model: we compute adjusted R^2 as $1 - (1 - \text{Nagelkerke's } R^2) \times (N-1) / (N-K-1)$, where N is the number of observations and K is the number of features. Categorical variables with M values

count as *M - 1* features, *e.g.,* student identity counts as 288 features. The low R^2 values here reflect the difficulty of predicting individual responses, compared to predicting aggregate performance averaged over many test items. Finally, classification accuracy is simply the percentage of cases for which the model correctly predicts whether the student got the cloze question right, *i.e.*, number of correct predictions / number of cloze items. We disaggregate by whether the student was right.

3.1. Is reading behavior informative?

Comparing the models in Table 2 shows how much reading behavior helps in predicting cloze outcomes. To penalize overfitting, we use adjusted R^2 to measure model fit. Comparing the full model to the ID+cloze model used in [9], we see that reading features uniquely explain 0.019 of the variance, increasing model fit by 10% relative (from 0.190 to 0.209). Comparing Reading-only and Cloze-only shows that reading features have almost as much explanatory power on their own as cloze features do (0.068 versus 0.070). In other words, listening to the student read (and get help) is roughly as informative as looking at the cloze question to predict cloze performance.

Comparing the fit of the ID+reading and ID-only models shows how much variance is uniquely explained by reading behavior, *i.e.*, not by student identity. To compute this non-overlapping portion of explained variance, we subtract the 0.134 adjusted R^2 for the ID-only model from the 0.175 adjusted R^2 for the ID+reading model. The result, 0.041, constitutes the bulk of the 0.068 adjusted R^2 explained by the reading behavior variables alone. Thus they capture something that student identity does not.

3.2. What construct do reading features measure, and how does it vary?

The analysis in Section 3.1 above shows that reading features pick up something. But what is this “something?” We hope that it’s local fluctuation of comprehension, yet it might be some other construct that we do not want to measure. To characterize the construct captured by our reading behavior variables, we analyze how it varies.

First, does this construct really vary over time, or not? That is, is it a state or a trait? There are at least two reasons to believe it’s a state. One reason is that the logistic regression model already includes student identity, which ought to capture any between-student differences that affect cloze performance – that is, student traits. Another reason is the relatively small overlap (0.027) in the variance explained by ID-only (0.134) and by Reading-only (0.068). We would expect almost complete overlap if the reading behavior variables just measured a student trait.

Second, does the construct fluctuate from one sentence to the next? Yes. We compared the “cloze sentence” Reading-only model to a variant that used reading features from the sentence just before the cloze question. We tested both models on a subset of 6901 cloze questions preceded by least three sentences in the same story. Adjusted R^2 was only 0.026 for the “previous sentence” model, versus 0.069 for the cloze sentence model. Reading factors uniquely explained only 0.018 of the variance, versus 0.042 in the cloze sentence model. The second and third sentences before the cloze question were even weaker predictors than the sentence just before the cloze question. Thus reading behavior on the cloze sentence itself predicts performance on the cloze question better than behavior on some other sentence.

Third, does the construct measure comprehension – or merely some artifact of answering the cloze question? For example, a student who gives a wrong cloze answer

expects a different sentence completion than the right one, and might read it differently (*e.g.*, less fluently) for that reason. We cannot entirely rule out this possibility. However, the fact that reading behavior on sentences preceding the cloze question is a significant (albeit weak) predictor of cloze performance provides hope that reading behavior on the completed cloze sentence also reflects comprehension.

4. Limitations and Future Work

We have used the rather loaded word "comprehension" to refer to the hidden state reflected by our reading behavior variables. The justification for using this word is that the state fluctuates from one sentence to the next, and explains variance in cloze performance not explained by the identity of the student or the difficulty of the cloze question. An alternative possibility is that the variables reflect some other state that affects cloze performance, such as student engagement in reading the story. However, engagement seems intuitively less likely than comprehension to fluctuate markedly from one sentence to the next, especially since a prior analysis of cloze behavior found that student engagement appeared relatively stable across a five-minute time scale [11].

Although we have shown that students' reading behavior variables explain variance in their cloze performance, explaining 7% of the variance is not enough for us to rely solely upon these reading features to detect comprehension fluctuations. Thus the work presented here is only a proof of concept, not a demonstration of feasibility. To make the method practical, it would be necessary to increase the model accuracy. The low R^2 may be attributed to errors in the speech recognition from which we derive some of our features, or to the inherently obscure relationship between reading and comprehension, which may be especially tenuous for more skilled readers. Possible solutions include exploiting additional prosodic clues such as accentuation and intonation, or combining reading features with other observations about the student.

However, perhaps the most glaring limitation of the current approach is the nature of the "training labels" that relate reading behavior to comprehension. A cloze question tests comprehension of a specific sentence – but it is a destructive test. Turning a sentence into a cloze question eliminates the opportunity to observe how the student would have read the sentence otherwise. The student's reading behavior might be scaffolded by seeing and hearing the cloze question first – or worse, affected differentially depending on whether the cloze answer is right or wrong.

Eliminating this limitation will require some other way to test comprehension of individual sentences. An obvious solution is to write comprehension questions by hand, and determine their difficulty empirically by administering them to a norming sample of students. However, this approach is labor-intensive. Project LISTEN uses multiple choice cloze questions because we can generate and score them automatically [2], and predict their difficulty [9]. The work reported here used the resulting data because it was available, not because it was ideal.

5. Conclusion

This paper is about analyzing students' reading behavior to detect fluctuations in their text comprehension automatically. We showed that machine-observable features of reading behavior improved a statistical model of students' performance on cloze

questions inserted in the text. Behavior on the completed cloze sentence was a stronger predictor than behavior on the sentences preceding the cloze question, suggesting that the reading behavior features are sensitive to fluctuations in comprehension.

This paper makes several contributions. First, we define the problem of detecting local fluctuations in text comprehension, which differs from the previously studied problem of assessing students' overall reading comprehension ability. Second, we propose a simple but useful conceptual model of the relationships among students' overall proficiency, reading behavior, fluctuations in text comprehension, and performance on cloze questions. Third, we translate this conceptual model into a statistical model by using Principal Components Analysis to derive useful predictors from system-specific features of reading behavior. Both the model and the methodology are potentially applicable to speech-enabled intelligent tutoring systems in other domains. Fourth, we evaluate the model on real data from Project LISTEN's Reading Tutor. Specifically, for this dataset, we show that behavioral indicators of comprehension predict student performance almost as well as question difficulty does.

This paper is a step toward future intelligent tutoring systems that detect comprehension fluctuations by listening to their students read aloud, thereby improving their estimates of what their students know. Such a capability could enhance the quality of tutorial decisions about what and how to teach.

Acknowledgements

This work was supported in part by the National Science Foundation under ITR/IERI Grant No. REC-0326153. Any opinions, findings, conclusions, or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the National Science Foundation or the official policies, either expressed or implied, of the sponsors or of the United States Government. We thank the educators, students, and LISTENers who generated and processed our data

References

1. Mostow, J. and G. Aist. Giving help and praise in a reading tutor with imperfect listening -- because automated speech recognition means never being able to say you're certain. *CALICO Journal*, 1999. *16*(3): p. 407-424.
2. Mostow, J., J. Beck, J. Bey, A. Cuneo, J. Sison, B. Tobin, and J. Valeri. Using automated questions to assess reading comprehension, vocabulary, and effects of tutorial interventions. *Technology, Instruction, Cognition and Learning*, 97-134. 2004.
3. Woodcock, R.W. *Woodcock Reading Mastery Tests - Revised (WRMT-R/NU)*. 1998, Circle Pines, Minnesota: American Guidance Service.
4. Beck, J.E., P. Jia, and J. Mostow. Automatically assessing oral reading fluency in a computer tutor that listens. *Technology, Instruction, Cognition and Learning*, 2004. *2*: p. 61-81.
5. Beck, J.E. and J. Sison. Using Knowledge Tracing in a Noisy Environment to Measure Student Reading Proficiencies. *International Journal of Artificial Intelligence in Education*, 2006. *16*(2): p. 129-143.
6. Berthold, A. and A. Jameson. Interpreting symptoms of cognitive load in speech input. *UM99, 7th International Conference on User Modelling*, 235-244. 1999. Wien, New York: Springer.
7. Müller, C., B. Groβmann-Hutter, A. Jameson, R. Rummer, and F. Wittig. Recognizing time pressure and cognitive load on the basis of speech: An experimental study. *UM 2001, 8th International Conference on User Modeling*, 24-33. 2001. Sonthofen, Germany.
8. Rayner, K., K.H. Chace, T.J. Slattery, and J. Ashby. Eye Movements as Reflections of Comprehension Processes in Reading. *Scientific Studies of Reading*, 2006. *10*(3): p. 241-256.
9. Hensler, B.S. and J. Beck. Better student assessing by finding difficulty factors in a fully automated comprehension measure. *8th International Conference on Intelligent Tutoring Systems*, 21-30. 2006. Jhongli, Taiwan.
10. Menard, S. Applied Logistic Regression Analysis. *Quantitative Applications in the Social Sciences*, 1995. *106*.
11. Beck, J. Engagement tracing: using response times to model student disengagement. *Proceedings of the 12th International Conference on Artificial Intelligence in Education (AIED 2005)*, 2005: p. 88-95.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Empowering researchers to detect interaction patterns in e-collaboration

Andreas Harrer [a,*], Rakheli Hever [b], Sabrina Ziebarth [a]
[a] *Universität Duisburg-Essen, Germany*
[b] *The Hebrew University of Jerusalem, Jerusalem, Israel*

Abstract. This paper describes an approach to support practitioners in the analysis of computer-supported learning processes by utilizing logfiles of learners' actions captured by the system. We enable researchers and teachers to identify and search for insightful patterns of the learning process in two ways: on the one hand patterns can be specified explicitly to be searched in logfiles; on the other hand automatic discovery of patterns that match configurable parameters is used to find most typical sequences in the logfiles. Both features have been realized in a stand-alone tool that accepts generic logfiles usable with a potentially wide variety of different learning systems. This is shown in an example of practical use of this tool in a project that supports researchers and moderators of graphical electronic discussions.

1. Introduction

CSCL-Systems (Computer Supported Collaborative Learning) provide a mediating function for collaboration, but recently also additional functionality for (pro-)active support of the collaboration for all people involved in CSCL, i.e. students, teachers, and researchers in learning processes. For this the system itself or additional tools have to incorporate computational techniques for analysis of the collaborative process to enable a better understanding of it. Some of these techniques consider the linguistic aspects of contributions, while others focus on activities on the domain level and the results of collaboration. The first line of research used either techniques for understanding of textual contributions [1] or classification of contributions on pragmatic level with semi-structured interfaces, such as sentence-openeners [2]. The second line investigated in analysis of domain-related actions in shared workspaces like analysis of the resulting states of collaboration [3] by checking structural properties. In CSCL-scenarios like classroom learning typically we cannot assume the system to be aware of the complete interaction, so that we tend to rely more on an understanding of actions and states on the domain-level of shared workspaces [4], taking also into account coordination acts, if available.

Learning Protocols are available in most CSCL-systems by logging the activities taking place in the collaborative environment. Logfiles are used in various ways [5,6] related to the AIED area. They have been used for manual inspection and interpretation [7], for exploration of mis-use of tutor support [8] and even for construction of tutors

*Correspondence to: Andreas Harrer, Abteilung für Informatik und angewandte Kognitionswissenschaft, Lotharstr. 63/65, 47057 Duisburg, Germany. Tel.: +49 203 379-3554; E-mail: harrer@collide.info

from real data [9]. In distributed architectures with a communication server these logfiles are produced automatically by logging the messages sent from learners' machines. Initially these logfiles are at a very low abstraction level and of little use for teachers, learners, and even the system to create an understanding of the ongoing collaboration.

In our collaborative applications the MatchMaker [10] communication server creates object-based logfiles, i.e. actions are related to the objects they operate on (e.g. creation, deletion, or modification of an object). These logfiles contain information about the user conducting the action, the original creator of the object and other aspects, that may be analysed for information about the ongoing collaboration. Even though this format is not as low-level as pure user-interface-events, the logfile itself is not useful for learners and teachers in its raw form. Our goal is to help users of CSCL systems to better understand the learning processes and to explore research questions. This is supported by providing a tool that enables the user to analyse logfiles from different CSCL systems.

2. Discovery of Patterns in Logfiles

Specific phases of collaboration or situations in the learning process, such as turn taking between learners or communication breakdowns, are indicated by typical sequences of user actions. The automatic discovery and visualization of these sequences helps teachers evaluating the process, researchers investigating their research questions, and students reflecting on their own activities. For that end, sequences have to be searched for matching a specific *pattern*; we will use the term pattern in this sense for a typical sequence of user actions. Depending on the type of educational application, logfiles of collaborative learning sessions contain usually an interspersed sequence of actions on coordination level, like chat messages, and domain-related actions, such as creating a UML-Class. This means that search for cohesive patterns (i.e. all actions directly succeeding each other) might not be sufficient: some interferences of actions, that are not relevant for the user, could be interspersed, so that a pattern can't be found in direct sequence. Thus the issue how to find patterns that are not strictly cohesive in the logfile has to be addressed in a tool supporting the user in logfile analysis. Another topic to consider when designing a tool for logfile analysis is that the researcher often has some hypothesis but cannot clearly specify how this hypothesis manifests itself on the low level of logfiles. For this it is desirable to automatically compute frequently occuring patterns that have not been specified before as a query. In the next subsections we will present our approach to tackle both the problems of dispersion of patterns and the extraction of unspecified patterns.

2.1. Explicit Specification of Patterns

Specified patterns can be found using standard algorithms or query engines if the patterns are cohesive in the logfile. Otherwise partial instances of patterns can be continued later in the logfiles. Because of this computational complexity we allow to filter out activities that are not relevant - if interspersed in a specified pattern - by choosing for each action type in the logfile if it is considered. The same can be done for specific users, e.g. when all actions of the teacher are filtered out to analyse just the learners' actions. By selecting actions in a suitable way incohesive patterns can be made cohesive for analysis purposes in the filtered logfile. To enable the user of our analysis approach to conveniently specify the patterns to be searched for, we provide three different methods for specification:

Rule specification by action types and rule composition: Based on the logfile to be analysed all action types (combination of objects and respective actions) that occur in the logfile are presented in a list. Thus actions, that do not occur in the logfile will not be part of the pattern to be searched. The user can specify a rule by selecting stepwise one or more of the action types in the order expected to occur in the log (see figure 1).

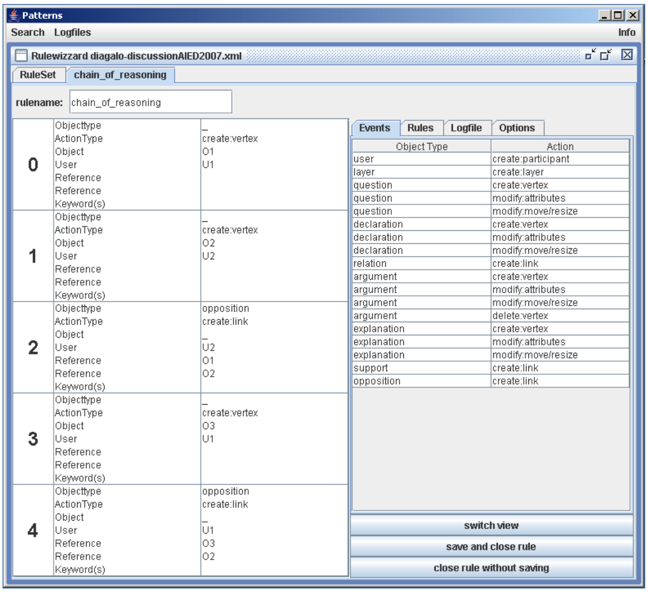


Figure 1. Rule specification by action types

Dependencies between the elements of a rule are specified by identical variables in the respective slot, e.g. in the User slot `U1`. The user can define rule sets and use already defined rules as subcomponents of new rules, to define complex rules by composition.

Specification by Example: The user also has the option to define a rule in a more convenient and informal way by selecting directly actions from the logfile as the pattern that is searched for. The tool automatically produces a rule by abstracting from the concrete values of the different slots of actions, but keeping the dependencies using the same variables if the values of slots were identical. Thus it is possible to review the logfile and use the system without any knowledge required about programming or rule specification.

Direct editing of rules: For the expert user there is also the option to directly edit and modify the rules (either created by rule specification or by example) at the level of the query language (cf. implementation section). This allows the user to add constraints or to the rule manually, but requires much more expertise than the previous methods, that hide the implementation level from the user who can define his rules on the action level.

Regardless of the method chosen for specification of rules the rules are compiled into rule sets that can be stored for later re-use. Thus a researcher loads a previously defined rule set, which compiles his standard rules for logfile analysis, chooses the rules from the set that should be actually used, and then conducts the pattern search (maybe after adding some specific rules for the current logfile/research question at hand).

2.2. Finding Unspecified Patterns

The much more challenging task of providing typical patterns that have not been specified before, demands a different algorithmic solution. For this we chose the so called "sawtooth" algorithm [11] that scans a string for the longest re-occuring substring starting from each position. This $O(n^2)$ algorithm iterates through a logfile and checks for each position in the log the maximal length of re-occurring sequences in the rest of the log. A completely un-supervised run will produce a lot of potential patterns of arbitrary length and frequency (minimum 2 to produce a re-occurrence at all); additionally this algorithm considers only directly succeeding patterns. To produce better results we provide several options for a user-guided pattern mining, i.e. the user defines general properties the patterns searched for have to comply to. These options are:

Objects to be considered: In a list the user can see all the objects used in the learning process captured in the logfile. For each item the user can inspect all the actions that have been conducted with this specific object. Based on this information she can choose if this object should be considered for the following pattern discovery run.

Object context: If the CSCL system producing the logfile represents relations between objects, as graphical modelling tools (e.g. Belvedere, CoLab, CoolModes, Digalo, ModellingSpace) do, an object's context is defined by its related objects. By configuring the maximal relational distance the size of the context can be varied; e.g. objects connected via a maximum path distance of 2 to the object can be selected for the search. This feature allows to detect patterns in specific regions of the learning products.

Users to be considered: Similar to the objects, a list of participants in the learning process is shown to the user of the pattern search tool. Choosing one participant in the list produces a list of actions conducted in the learning process by the participant. By reviewing these the user can decide which users should be considered for the search.

Action types to be considered: This globally allows or disallows action types associated with a specific object type in the pattern discovery. E.g. the user can decide, that moving a UML-Class object is not relevant for her research question who typically worked on content-level, but changing attributes of a UML-Class object is relevant.

Pattern length and Frequency: These options constrain the number of patterns to the ones that comply to minimum lengths of pattern and minimum frequency of hits in the logfile. Only the compliant ones are proposed to the user as candidate patterns.

By combining several of these options the user guides the pattern discovery process in the direction of the patterns he is interested in. The pattern discovery produces rules, each characterizing one pattern schema fulfilling all the option's criteria, which can be used as if they have been specified by the user directly (cf. previous subsection).

3. Tool Implementation

This section describes our practical implementation of the previously explained concepts. It is developed as a stand-alone analysis tool that is flexibly usable by analysts of learning processes with diverse learning systems. Important issues are:

Generic logfile format: To support a wide variety of different learning systems we defined a format that uses standardized attributes required for our analysis process (e.g. action type, user, timestamp), yet on the other hand is able to preserve and use

tool-specific information (e.g. for a graph-based tool relations between objects) in the analysis. The chosen logfile format is a XML representation of events triggered by the learner, so that other XML-based logfiles can be converted conveniently to this format.

Prolog runtime system as query engine: The patterns to be searched for can be considered as rules, especially when composing more complex patterns based on existing ones; this maps well to using a rule within a composition rule. Instead of implementing a search engine from scratch we explored the option of using existing rule-based engines for logical programming. Because of our use of the Java language for the graphical user interface and the logfile parsing, JESS (Java Expert System Shell) and SWI-Prolog were the main options: both provide interoperability with Java applications and offer optimized search engines. In practical tests we experienced performance problems with JESS and thus chose SWI-Prolog in conjunction with the JPL Java Interface.

Conducting the pattern search is a two-step procedure: first for every event in the logfile a Prolog fact is defined according to the general format:

```
event(eventid, objectid, action, objecttype,
      timestamp, user, additionalAttributes)
```

A pattern is described internally as a sequence containing event structures with variable elements and (for composition rules) subrules. The elements of an event structure are defined schematically: *user-defined values* are used as concrete values in the rule, *user-defined variables* are used in the rule consistently for each occurrence of the variable, and *elements the user didn't specify* are used as the unspecified Prolog variable "_". When composing rules, the representation of the subrule is added to the rule definition.

Implementation of the Sawtooth-Algorithm Due to the generic input logfile we need a generally applicable pattern search algorithm without domain-dependent optimizations. Our implementation of the Sawtooth algorithm operates on a sequence of action events: a section of the logfile is matched against the logfile using the *patternlength* and *frequency* options. If the section satisfies the two options, it is marked as an occurrence of a pattern and is expanded with additional subsequent actions in the next iteration; otherwise a new section will be tested. At termination the algorithm has marked possible patterns and their occurrence. To generate a suitable assignment of parameter values that represents the dependencies between different actions (such as same user, same object) an incremental processing similar to the *apriori* algorithm is conducted.

Visualization of hits: Found patterns are displayed in a diagram illustrating the timeline of the learning session. Every point of the graph is an action event within a pattern and is connected with its previous and succeeding action events. Every point has a tooltip window containing all information of its action event. Additionally all hits are arranged in a tree-structure: the tree allows the user to see an abstract representaion of the rules belonging to specific patterns and the parameter values of specific hits. Selecting one rule respective one pattern highlights the corresponding subgraph in the diagram.

4. Usage as a Research Tool for Discussion Analysis in ARGUNAUT

The approach presented has been implemented within a tool described in [12] with early results of the feasibility of the proposal based on research data from a case study [9]. Since then it was used in empirical research in the EU-funded Argunaut project (IST contract no. 027728). The project's goal is to provide moderators with computer-based

tools to support and increase their effectiveness and thereby the quality of monitored e-discussions. ARGUNAUT aims at delivering a unified mechanism of awareness and feedback to support moderators in multiple e-discussion environments.

One of the specific aspects of the Argunaut project is that the moderation support is not constrained to one dedicated discussion environment; versatile environments, currently the Digalo, FreeStyler, and Cool Modes applications[1] developed by different partners, can make use of the analysis and moderation features of the system. Thus, while initially the Pattern Discovery Tool was developed for the collaborative workspace environments FreeStyler & Cool Modes, the provision of a standardised XML-based logging format enables the tool to analyse all logfiles compliant to this format.

The pattern discovery tool was used as a research tool to help moderators or researchers identify relevant discussion patterns that occured in real discussions with visual graph-based languages (see 2). After a first practical use of the tool with real data and feedback from the practitioners, we added the functionality to search for texts (e.g. keywords) in the discussion cards; this enables us to combine the process-oriented dimension of actions on graphical discussion objects with content-based textual aspects, such as characteristic phrases for agreement, disagreement, explanation etc. We allows to utilize the diverse background of the project partners in pedagogy, linguistics, argumentation and interaction analysis in a combined analysis.

The pedagogical experts in the project then used the option of explicitly specifying patterns of interest by action types. The complexity and abstraction level of rules varied widely between relatively directly observable phenomena, such as *agreement*, to quite abstract and complex concepts, such as *change of opinion*. A selection of these rules is:

Agreement: a sequence in which two discussion shapes are created by different users and a "support" link is added between them.

Chain of reasoning (in response to attack): a sequence in which a user creates a discussion shape, another user creates a shape linked to the first shape using an "opposition" link, and the first user responds to this opposition by creating a shape and linking it to the other user's shape with an "opposition" link. A specification of this pattern can be seen in figure 1, while figure 2 shows 2 hits of the pattern manifested in a Digalo map.

Possible change of opinion: a sequence in which different users create two discussion shapes and a link between them. After this the same users create two discussion shapes with a different link (e.g. opposition becomes support or vice versa).

Revision / deletion: a sequence in which a user first creates a discussion object and later this object is modified or deleted. Discussion objects created by a specific user may be modified/deleted by the same user who created them or by other users, and this distinction (expressed by different sub-rules) is meaningful to the moderator or researcher.

Reasoned opposition: a sequence in which two discussion shapes are created by different users and an "opposition" link is added between them. The second shape is specified to contain reasoning keywords, such as "because", "example", etc. (shape types may also be specified).

Question and explanation: a sequence in which a user creates a discussion shape, another user creates a discussion shape of the "question" type (and/or one containing questioning keywords) which is later linked to the first shape. In the following steps, the first user responds to this question by creating another shape of the "explanation" or

[1] http://dunes.gr , http://www.collide.info/downloads

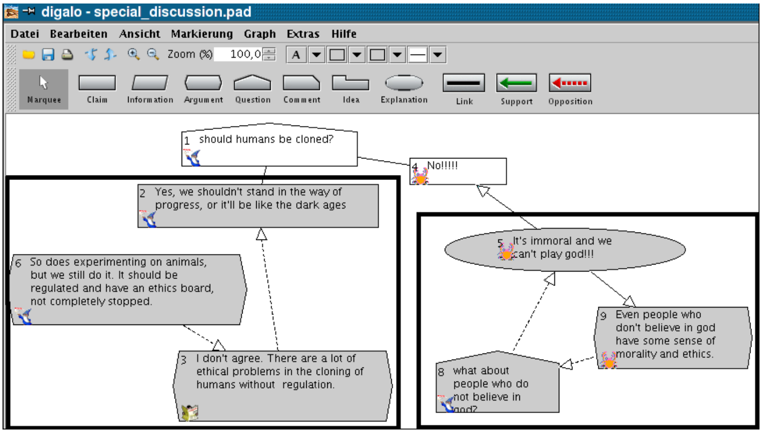


Figure 2. Digalo discussion map with two identifed hits of the chain of reasoning pattern

"reason" type (and/or a shape containing reasoning/explanation keywords) and linking it to the other user's shape.

The experts' approach in the ARGUNAUT project was to pre-define possible action sequences thought to be meaningful, i.e. representing specific theoretical or pedagogical phenomena (examples above) and to look for such sequences. The rules specified were applied to 582 XML log files of graphical discussion maps (created by small groups of school pupils with the Digalo environment). When applying a ruleset to a complete folder of logfiles, the results are shown in an ordered list containing the logfiles with most hits on top and then descreasing number of hits. This helps to give a first overview about the logfiles and the characteristics that are expressed by the numbers of each rule's hits. Typically a researcher would define her/ his own ruleset, apply it to the whole dataset and thus obtain a list of candidate XMLs (the discussions which show the largest number of hits) for deeper investigation at the fine-grained logfile level: then the temporal sequence, the actors involved in the hits can be inspected more closely.

The practical usage by the experts showed that they were capable of using the tool for specification of quite complex phenomena relevant to e-discussions. Yet, we also found out that the hits found by a discovery run sometimes were not considered as useful. This is due to over-generalization by the experts, which could have been avoided by applying the constraints and options offered in the tool. An example of this is that a rule for *repeated modification* of an object, that was specified with 3 modification actions, produced several hits for 4 modification events (all three-sets possible); this could have been avoided using options for time intervals or interspersed actions within the rule. To adress the issue of non-appropriate hits we added a feature to annotate the hits according to whether they truly represent the semantic interpretation of the expert ('good hits'), and save this information in XML format for further reference. Another observation from this experimentation is, that the experts frequently tried to define phenomena with structural aspects (e.g. shapes linked to 3 other objects) without the temporal dimension the tool allows for. Thus, while from these experiments of usage we can conclude that the tool is usable and useful, there is also a challenge in operationalizing the experts' intentions as rules. We assume that this requires close collaboration between the experts in the domain with people of formal background to clearly specify these rules.

5. Conclusion and Outlook

This paper presented our approach to support analysts of learning processes, i.e. researchers and teachers, in understanding the processes captured in computer-created logfiles. We provided functionality to find typical patterns of learning actions that can be either specified by the user via a convenient graphical interface or that are discovered automatically with configuration options for the user. We also adressed the issue, that especially in logfiles of collaborative processes, patterns might be interspersed with actions that are not so relevant for the analyst, but which inhibit the finding of meaningful patterns. This concept has been implemented within a tool and put to practical use in a European funded project with logfiles created by usage of a graphical discussion environment of one of our project partners. The analyis tool is designed to accept not only these logfiles, but provides a generic logfile format and suitable XSL-T transformation scripts to map the logfiles of arbitrary learning systems to the generic format. We will use our tool for analysis of lab and classroom experiments previously done manually [9].

Acknowledgements Our thanks go to Michael Vetter, Jens Brauckmann, Stefan Thür for their work on the pattern tool and our Argunaut partners for their comments.

References

[1] Carolyn P. Rose, Andy Gaydos, Brian S. Hall, Antonio Roque, and Kurt VanLehn. Overcoming the knowledge engineering bottleneck for understanding student language input. In *Proc. of AIED 2003*, pages 315–322, Amsterdam, 2003. IOS Press.

[2] Michael Baker and Kristine Lund. Flexibly structuring the interaction in a CSCL environment. In *Proc. of EuroAIED'96*, Lissabon, 1996.

[3] Maria de los Angeles Constantino-Gonzalez and Daniel Suthers. A coaches collaborative learning environment for entity-relationship modelling. In *Proc. of ITS 2000*, pages 325–333. Springer, 2000.

[4] Katrin Gaßner, Marc Jansen, Andreas Harrer, Kai Herrmann, and Ulrich Hoppe. Analysis methods for collaborative models and activities. In *Proc. of CSCL 2003*, pages 369–377. Kluwer, 2003.

[5] Cecily Heiner, Ryan Baker and Kalina Yacef (eds.). *Proc. of ITS2006 Workshop on Educational Data Mining*, Jhongli, Taiwan, 2006.

[6] Joseph E. Beck, Esma Aimeur, and Tiffany Barnes (eds.). *Educational Data Mining: Papers from the 2006 AAAI Workshop, Technical Report WS-06-05.* Menlo Park, California, 2006.

[7] Erica De Vries, Kristine Lund, and Michael J. Baker. Computer-mediated epistemic dialogue: Explanation and argumentation as vehicles for understanding scientific notions. *The Journal of Learning Sciences*, 11(1): 63–103, 2002.

[8] Ryan S. Baker, Albert T. Corbett, and Kenneth R. Koedinger. Detecting student misuse of intelligent tutoring systems. In *Proc. of ITS2004*, 2004.

[9] Andreas Harrer, Bruce M. McLaren, Erin Walker, Lars Bollen, and Jonathan Sewall. Creating cognitive tutors for collaborative learning: Steps toward realization. *User Modeling and User-Adapted Interaction: The Journal of Personalization Research*, 16: 175–209, 2006.

[10] Marc Jansen. Matchmaker - a framework to support collaborative java applications. In *Proc. of AIED 2003*, pages 529–530, Amsterdam, 2003. IOS Press.

[11] Heinz Ulrich Hoppe and Rolf Plötzner. Inductive knowledge acquisition for a unix coach. In *Informatics and Psychology Workshop*, pages 313–335, 1989.

[12] Andreas Harrer, Michael Vetter, Stefan Thür, and Jens Brauckmann. Discovery of patterns in learner actions. In *Proc. of AIED 2005*, pages 816 – 818, Amsterdam, 2005. IOS Press.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Is Over Practice Necessary? – Improving Learning Efficiency with the Cognitive Tutor through Educational Data Mining

Hao CEN, Kenneth R. KOEDINGER and Brian JUNKER
Carnegie Mellon University
5000 Forbes, Pittsburgh, PA, U.S.A.
hcen@andrew.cmu.edu, koedinger@cmu.edu, brian@stat.cmu.edu

Abstract. This study examined the effectiveness of an educational data mining method – Learning Factors Analysis (LFA) – on improving the learning efficiency in the Cognitive Tutor curriculum. LFA uses a statistical model to predict how students perform in each practice of a knowledge component (KC), and identifies over-practiced or under-practiced KCs. By using the LFA findings on the Cognitive Tutor geometry curriculum, we optimized the curriculum with the goal of improving student learning efficiency. With a control group design, we analyzed the learning performance and the learning time of high school students participating in the Optimized Cognitive Tutor geometry curriculum. Results were compared to students participating in the traditional Cognitive Tutor geometry curriculum. Analyses indicated that students in the optimized condition saved a significant amount of time in the optimized curriculum units, compared with the time spent by the control group. There was no significant difference in the learning performance of the two groups in either an immediate post test or a two-week-later retention test. Findings support the use of this data mining technique to improve learning efficiency with other computer-tutor-based curricula.

Keywords: Data mining, intelligent tutoring systems, learning efficiency

Introduction

Much intelligent tutoring system (ITS) research has been focused on designing new features to improve learning gains measured by the difference between pre and post test scores. However, learning time is another principal measure in the summative evaluation of an ITS. Intelligent tutors contribute more to education when they accelerate learning [9]. Bloom's "Two Sigma" effect of a model human tutor [4] has been one of the ultimate goals for most intelligent tutors to achieve. So should be the "Accelerated Learning" effect shown by SHERLOCK's offering four-year's trouble shooting experience in the space of seven days of practice [12].

Cognitive Tutors are an ITS based on cognitive psychology results [11]. Students spend about 40% of their class time using the software. The software is built on cognitive models, which represent the knowledge a student might possess about a given subject. The software assesses students' knowledge step by step and presents curricula tailored to individual skill levels [11]. According to Carnegie Learning Inc., by 2006, Cognitive Tutors have been widely used in over 1300 school districts in the U.S. by

over 475,000 secondary school students. With such a large user base, the learning efficiency with the Tutor is of great importance. If every student saves four hours of learning over one year, nearly two million hours will be saved. To ensure adequate yearly progress, many schools are calling for an increase in instructional time. However, the reality is that students have a limited amount of total learning time, and teachers have limited amount of instructional time. Saving one hour of learning time can be better than increasing one hour of instructional time because it does not increase students' or teachers' work load. Moreover, if these saved hours are devoted to other time-consuming subjects, they can improve the learning gains in those subjects.

Educational data mining is an emerging area, which provides many potential insights that may improve education theory and learning outcomes. Much educational data mining to date has stopped at the point of yielding new insights, but has not yet come full circle to show how such insights can yield a better intelligent tutoring system (ITS) that can improve student learning [2, 3].

Learning Factors Analysis (LFA) [6, 5] is a data-mining method for evaluating cognitive models and analyzing student-tutor log data. Combining a statistical model [10], human expertise and a combinatorial search, LFA is able to measure the difficulty and the learning rates of knowledge components (KC), predict student performance in each KC practice, identify over-practiced or under-practiced KCs, and discover "hidden" KCs interpretable to humans. The statistical model is shown in Eq. (1).

$$\log\left(\frac{P_{ijt}}{1-P_{ijt}}\right) = \sum \theta_i X_i + \sum \beta_j Y_j + \sum \gamma_j Y_j T_{jt} \quad (1)$$

P_{ijt} is the probability of getting a step in a tutoring question right by the ith student's tth opportunity to practice the jth KC. The model says that the log odds of P_{ijt} is proportional to the overall "smarts" of that student (θ_i) plus the "easiness" of that KC (β_j) plus the amount gained (γ_j) for each practice opportunity. With this model, we can show the learning growth of students at any current or past moment.

By applying LFA to the student log data from the Area unit of the 1997 Geometry Cognitive Tutor, we found two interesting phenomena. On the one hand, some easy (i.e. high β_j) KCs with low learning rates (i.e. low γ_j) are practiced many times. Few improvements can be made in the later stages of those practices. KC rectangle-area is an example. This KC characterizes the skill of finding the area of a rectangle, given the base and height. As shown in Figure 1, students have an initial error rate around 12%. After 18 times of practice, the error rate reduces to only 8%. The average number of practices per student is 10. Many practices spent on an easy skill are not a good use of student time. Reducing the amount of practice for this skill may save student time without compromising their performance. Other over-practiced KCs include square-area, and parallelogram-area. On the other hand, some difficult (i.e. low β_j) KCs with high learning rates (i.e. high γ_j) do not receive enough practice. Trapezoid-area is such an example in the unit. But students received up to a maximum of 6 practices. Its initial error rate is 76%. By the end of the 6th practice the error rate remains as high as 40%, far from the level of mastery. More practice on this KC is needed for students to reach mastery. Other under-practiced KCs include pentagon-area and triangle-area.

Having students practice less than needed is clearly undesirable in the curriculum. Is over practice necessary? The old idiom "practice makes perfect" suggests that the

more practice we do on a skill, the better we can apply the skill. Many teachers believe that giving students more practice problems is beneficial and "would like to have the students work on more practice problems", even when "[students] were not making any mistakes and were progressing through the tutor quickly"[7].

We believe that if the teachers want more problems for their students to practice unmastered KCs or useful KCs not covered by the curriculum, more practice is necessary. To support KC long-term retention, more practice is necessary but needs to be spread on an optimal schedule [1, 14]. In the rectangle-area example, where all the practice for this KC is allocated in a short period, more practice becomes over practice, which is unnecessary after the KC is mastered.

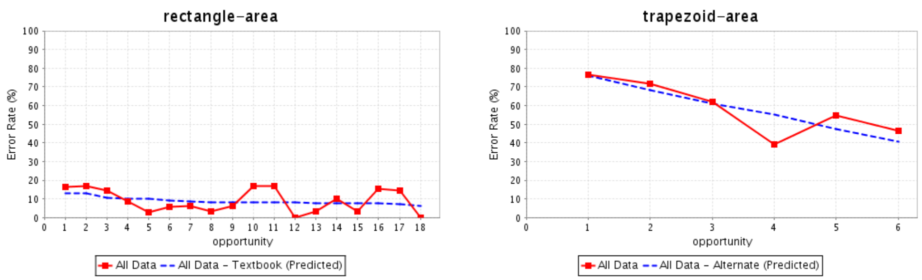


Figure 1 Learning Curve of Rectangle-Area and Trapezoid-Area – The solid lines are the actual error rates over the ordered number of practices. The dotted lines are the error rates predicted by LFA.

1. Design of the Optimized Tutor

What caused the over practice in the Cognitive Tutor curriculum? Cognitive Tutor uses the Knowledge Tracing algorithm to update its estimates of students' mastery of KCs [8]. Based on these estimates, the Tutor chooses to give students the problems with the skills students need to practice more. Table 1 explains the meaning of the four parameters $P(L_0)$, P (T), P(Guess), P(Slip) used in the update. We discovered that the 1997 Tutor used the same set of parameter estimates for all the KCs, as shown in Table 1 column 3. Then we fit the four parameters for each KC with the student log data and used the fit parameters to estimate the amount of practice for these KCs in the same dataset. We found that 58% out of 4102 practices and 31% of 636 exercise questions were done after the students had reached mastery. On the other hand, 119 more practices were needed for all students to master those under-practiced KCs. Although applying the fit parameter estimates on the training data may incur over fitting, the finding does suggest that using a set of carefully calibrated parameter estimates may improve learning efficiency.

To test the effect of calibrated Knowledge Tracing parameters, we planned a study in 2006, when the Geometry Cognitive Tutor had evolved into its 2006 version. The 2006 Tutor breaks the single 1997 area unit into 6 area units (Squares & Rectangles, Parallelograms, Triangles, Trapezoids, Polygons, and Circles), and has a different cognitive model, curriculum design, interface, and student population from its predecessor.

Table 1 Knowledge tracing parameters used in the 1997 Cognitive Geometry Tutor

Parameter	Meaning (The probability that …)	Estimate
$P(L_0)$	the KC is initially known	0.25
P(T)	the KC transit form an unknown state to a known state	0.2
P(Guess)	a student will apply a KC correctly even if the KC is not learned	0.2
p(Slip)	a student will apply a KC incorrectly even if the KC is learned	0.1

53 KCs are specified in the six units. 32 of them involve calculating perimeters and areas of various shapes. 21 of them involve extracting specific numbers from the text questions and entering them into the tutor interface. The numbers of KCs in each of the 6 units are 19, 7, 7, 8, 4, and 6 respectively. The same set of knowledge tracing parameter estimates shown in Table 1 were used for all its KCs.

Because the 2006 Tutor was just released several weeks before our study, there were no student-log data available from that Tutor to evaluate the KCs empirically. The most recent data available were from the 2005 version of the Cognitive Geometry Tutor, which used the same set of knowledge tracing parameter estimates shown in Table 1. Not surprisingly, we found over-practice and under-practice in that tutor. The over-practiced KCs and the under-practiced KCs are slightly different across the tutors.

Because the cognitive model in the 2005 Tutor is slightly different from the model in the 2006 Tutor, we could not exactly copy those parameter estimates into the 2006 Tutor. To approximate the 2006 estimates, we grouped the KCs in the 2006 Tutor into several homogeneous groups according to their degrees of over-practice in 2005. This is a qualitative mapping of the parameters of one tutor version with a different student population to the parameters of another tutor version. Within each group, KCs share the same parameter estimates. Because we had no relevant information on slips or guesses, we mainly focused on adjusting $P(L_0)$ and P(T) in our study. Between groups, KCs vary mainly on $P(L_0)$. If a KC shows a much higher learning rate P(T) than the original parameter estimate .2, we increased P(T) for that KC. Meanwhile, in order to reduce the danger of under-practice, we reduced all the $P(L_0)$ by a certain amount from the fit estimates. The final parameter estimates in the optimized tutor have the following changes.

- The under-practiced KC (circle-area) is set to a lower $P(L_0) = 0.2$.
- The under-practiced KC with a high learning rate (triangle-area) is set to a lower $P(L_0) = 0.2$, and a higher $P(T) = .5$.
- The slightly-over-practiced KCs (circle-circumference, trapezoid-area, trapezoid-perimeter, triangle-perimeter) are set to $P(L_0) = 0.5$.
- The moderately-over-practiced KCs (parallelogram-area, parallelogram-perimeter, rectangle-area, rectangle-length-or-width, rectangle-perimeter, square-area, square-perimeter, square-side-length) are set to $P(L_0) = 0.7$.
- All the KCs for information extraction are set to $P(L_0) = 0.9$.

Based on these changes, their locations in the units, and the number of KCs in each unit shown, we made the following research hypothesis. Compared with the students in the control condition, students using the optimized tutor will learn the same amount but spend

- much less time in unit 1 and 2 (Squares and Parallelograms);
- moderately less time in unit 3 (Triangles);

- the same amount of time in unit 4 and 5 (Trapezoids and Polygons)
- more time in unit 6 (Circles)

2. Design of the Study

The experiment was conducted in the regular instruction of the geometry course. 110 students from a total of 6 classes in a high school near Pittsburgh participated in the study. They were all taught by the same teacher. The students were randomly assigned to the optimized condition, where they would be using the optimized tutor, or to a control condition where they would be using the original 2006 Tutor with no modifications. We designed a pretest, a post test and a retention test to assess students' ability to solve geometry area and perimeter problems. The test forms were counter-balanced. Each test has 13 – 14 test items including both regular and transfer items.

In both conditions, the students took the pretest shortly before they started working on the first unit. Then students spent about 3 days per week on regular classroom instruction and 2 days per week using the Tutor in a computer lab. In the lab, students worked on either the optimized tutor or the original Tutor, according to the condition they had been assigned to. Shortly after finishing the 6th unit, they took the post test. In two weeks after each student finished the post test, we gave each student a retention test. Students' interaction and learning time with the tutor were logged. Due to student attrition and recording errors, 94 students took the pretest; 84 students took the post test; 64 students took the retention test; 73 students took both the pretest and the post test and had valid test scores; and 62 students had valid log data.

3. Results

We found that the optimized group learned as much as the control group but in less time. As seen in Figure 2 (left), the two groups have similar scores in both the pre test and the post test. The amount of learning gain in both groups is approximately 5 points. To further examine the treatment effect, we ran an ANCOVA on the post test scores, with condition as a between subject factor, the pretest scores as a covariate, and an interaction term between the pretest scores and the condition. The post test scores are significantly higher than the pretest, $p < .01$, suggesting that the curriculum overall is effective. Meanwhile neither the condition nor the interaction are significant, $p = 0.772$, and $p = 0.56$ respectively. As shown in the Figure 2 (right), we found no significant difference in the retention test scores ($p = 0.602$, two tailed). The results from the post test and the retention tests suggest that there is no significant difference between the two groups on either of the two tests. Thus, over practice does not lead to a significantly higher learning gain.

The actual learning time in each unit matches our hypotheses. As shown in Table 2, the students in the optimized condition spent less time than the students in the control condition in all the units except in the circle unit. The optimized group saved the most amount of time, 14 minutes, in unit 1 with marginal significance $p = .19$; 5 minutes in unit 2, $p = .01$, and 1.92, 0.49, 0.28 minutes in unit 3, 4, and 5 respectively. In unit 6, where we lowered $P(L_0)$, the optimized group spent 0.3 more minutes. Notice the percentage of the time saved in each unit. The students saved 30% of tutoring time in

unit 2 Parallelogram, and 14% in unit 1 Square. In total students in the optimized condition saved around 22 minutes, an 12% reduction in the total tutoring time.

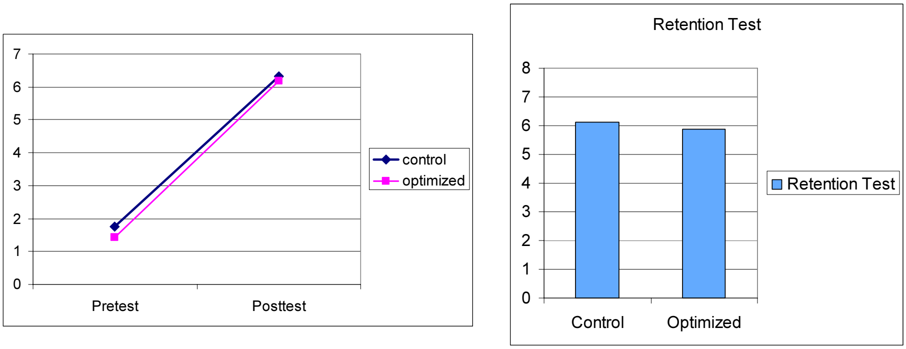


Figure 2 Pretest and post test scores over the two conditions (left) and the retention test scores (right)

Table 2 Time cost in the six tutor curriculum units. The time is in minutes.

	Optimized	Control	Time saved	% time saved	t Stat	P(T<=t) one-tail
Square	87.16	101.18	14.02	14%	-0.89	0.19
Parallelogram	11.83	16.95	5.12	30%	-2.58	0.01
Triangle	13.03	14.95	1.92	13%	-0.91	0.18
Trapezoid	26.39	26.88	0.49	2%	-0.15	0.44
Polygon	10.58	10.86	0.28	3%	-0.18	0.43
Circle	13.42	13.12	-0.30	-2%	0.18	0.43
Total	162.41	183.93	21.52	12%		

4. Conclusion and Generalizing this Method for Other Tutors

In this study, we presented a method using educational data mining to identify the inefficiency in an ITS. Rather than concluding the research just with the data mining findings, we used the findings to optimize the tutor and showed that students using the optimized tutor saved a significant amount of time compared with the time spent by the control group while both groups achieved a similar learning outcome. While prior studies have demonstrated benefits of cognitive mastery and knowledge tracing, this study is the first we know of that shows using tuned parameters in the tutor yields better learning than hand-set parameters. The knowledge tracing parameters in the existing Cognitive Tutor are hand set.

Although the data was from an earlier version of the tutor with a different student population from that in this study, the fact that efficiency gains were found suggest that this method has potential for generalizing across student population and tutor versions. In addition, this method works for any ITS where student outcomes are labeled with knowledge components and the opportunity to use those KCs. For instance, this method

could apply in any other model-tracing tutor or constraint-based tutors [13], where the constraints are the KCs.

5. Other Inefficiencies Found and Future Work

We also found other interesting learning inefficiencies through educational data mining. One tutoring question in the first unit Square involves 4 KCs – Find rectangle length or width from given area, Find rectangle length or width from given perimeter, Find rectangle area, and Find rectangle perimeter. In Cognitive Tutor, each question is presented as an indivisible piece. What if a student has mastered KC 1, 2, 3 but has not mastered KC 4? Unfortunately, the student still has to answer the whole question and practice all four KCs. One way to tackle this inefficiency is to author more tutoring questions, each targeting one specific KC. The tutor would select the corresponding question for an unmastered KC. In the example above, we would create four questions with one for each KC and the tutor would select a question requiring only KC 4. Another solution would be to have the Tutor automatically solve the steps for the mastered KCs and only ask the student to solve the steps for unmastered KCs.

The third and the forth inefficiency relates to cognitive modeling. In the Learning Factors Analysis study [6], we showed that some KCs are better merged into one KC. For example, if we merge "finding the circumference of a circle given diameter (circle-circumference)" and "finding the diameter of a circle given circumference (circle-diameter)" into a single KC circle-CD, the cognitive model fits the log data better and has less complexity. This suggests that as students become more familiar with some low level KCs, the instruction may need to target their aggregated counterparts. Sharp-eye readers may have noticed that unit 1 Square and Rectangle took over 100 minutes in the control condition, greater than the total time of the remaining five units. One cause of this large time consumption is that unit 1 has 19 KCs, over one-third of the total number of KCs in the six units. Due to the similarity between squares and rectangles, we hypothesized that students may only need one representation for both shapes. For example, "square-area" and "rectangle area" may be well represented with one skill. If the hypothesis is true, we can reduce the number of skills in this unit by half. We plan to use LFA to determine appropriateness of merging these skills in a future study.

By extending LFA, Rafferty and Yudelson [15] found that students have different cognitive models -- low-achieving students have more elaborate cognitive models while high-achieving students have more compact models. High-achieving students may do well with even less practice if knowledge tracing uses a compact model for them. To tackle this inefficiency, we need to design a set of representative cognitive models for different groups of students. High-achieving students use a compact model so that they can go through the curriculum more quickly while low-achieving students may be assigned an elaborate model so that they receive more detailed practice. The ambitious solutions to the last two inefficiencies reflect an ITS design principle "adjust the grain size of instruction with learning" [9].

The methods we have discussed have various strengths and implementation difficulties. The method we presented in this paper -- using a set of carefully calibrated knowledge tracing parameter estimates -- can improve learning efficiency without incurring any major tutor modification. It can be easily scaled to other tutor curricula.

The findings support the use of this data mining technique to improve learning efficiency with ITS curricula.

Acknowledgements

This research is sponsored by a National Science Foundation grant SBE-0354420 to the Pittsburgh Science of Learning Center. We thank Carnegie Learning for generously providing the Cognitive Geometry Tutor for our study. We thank Steve Ritter, Joseph Beck, and Albert Colbert for their comments on earlier versions of this paper. We thank Jonathan Steinhart, Christy McGuire, and Frank Baker for their support in tutor modifications and data collection. We thank Alida Skogsholm and Ben Billings for their work on Datashop (http://www.learnlab.org/technologies/datashop), a central repository for educational research data and a set of rich analysis and reporting tools. We thank Colleen Conko for supporting us to use her classes as the study environment. We thank Megan Voorhis for her writing support.

References

[1] J. R. Anderson, J. M. Fincham and S. A. Douglass, *Practice and retention: A unifying analysis* Journal of Experimental Psychology: Learning, Memory, and Cognition, 25 (1999), pp. 1120-1136.
[2] T. Barnes, *The Q-matrix Method: Mining Student Response Data for Knowledge*, *American Association for Artificial Intelligence 2005 Educational Data Mining Workshop*, 2005.
[3] J. E. Beck and B. P. Woolf, *Evaluating tutorial actions with a user model*, *User Modeling and User Adapted Interaction*, 2007.
[4] B. S. Bloom, *The 2 Sigma Problem: The Search for Methods of Group Instruction as Effective as One-to-One Tutoring*, Educational Researcher, 13 (1984), pp. 4-16
[5] H. Cen, K. Koedinger and B. Junker, *Automating cognitive model improvement by a* search and logistic regression*, *American Association for Artificial Intelligence 2005 Workshop on Educational Datamining*, 2005.
[6] H. Cen, K. Koedinger and B. Junker, Learning Factors Analysis - A General Method for Cognitive Model Evaluation and Improvement, 8th International Conference on Intelligent Tutoring Systems, 2006.
[7] J. Cital, *A business email report to Carnegie Learning Inc*, 2006, pp. Email.
[8] A. T. Corbett and J. R. Anderson, *Knowledge tracing: Modeling the acquisition of procedural knowledge*, *User Modeling and User-Adapted Interaction*, 1995, pp. 253-278.
[9] A. T. Corbett, K. Koedinger and J. R. Anderson, *Intelligent Tutorins Systems*, in M. G. Helander, T. K. Landauer and P. Prabhu, eds., *Handbook of Human-Computer Interaction*, Elsevier Science, Amsterdam, The Netherlands, 1997.
[10] K. Draney, P. Pirolli and M. Wilson, *A Measurement Model for a Complex Cognitive Skill*, *Cognitively Diagnostic Assessment*, Erlbaum, Hillsdale, NJ 1995.
[11] K. R. Koedinger and A. T. Corbett, *Cognitive Tutors: Technology Bringing Learning Science to the Classroom*, in K. Sawyer, ed., *The Cambridge Handbook of the Learning Sciences*, Cambridge University Press., 2006, pp. 61-78.
[12] A. Lesgold, S. Lajoie, M. Bunzo and G. Eggan, *Sherlock: A Coached Practice Environment for an Electronics Troubleshooting Job*, in J. H. Larkin and R. W. Chabay, eds., *Computer-assisted instruction and intelligent tutoring systems: shared goals and complementary approaches*, Lawrence Erlbaum Associates, 1988.
[13] A. Mitrovic, M. Mayo, P. Suraweera and B. Martin, *Constraint-based tutors: a success story*, *14th Int. Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems IEA/AIE-2001*, Springer-Verlag Berlin Heidelberg, Budapest, 2001.
[14] P. I. Pavlik and J. R. Anderson, *Practice and Forgetting Effects on Vocabulary Memory: An Activation Based Model of the Spacing Effect* Cognitive Science, 29 (2005), pp. 559-586.
[15] A. Rafferty and M. Yudelson, *Applying LFA for Building Stereotypical Student Models*, *PSLC Summer School 2006 Projects*, Pittsbugh Science of Learning Center, Pittsburgh PA, 2006.

Posters

This page intentionally left blank

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Benefits of Handwritten Input for Students Learning Algebra Equation Solving

Lisa ANTHONY, Jie YANG and Kenneth R. KOEDINGER
Human-Computer Interaction Institute, Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh PA 15213

Abstract. Building on past results establishing a benefit for using handwriting when entering mathematics on the computer, we hypothesize that handwriting as an input modality may be able to provide significant advantages over typing in the mathematics learning domain. We report the results of a study in which middle and high school students used a software tutor for algebra equation solving with either typing or handwriting as the input modality. We found that handwriting resulted in similar learning gains in much less time than typing. We also found students seem to experience a higher degree of transfer in handwriting than in typing based on performance during training. This implies that students could achieve farther goals in an intelligent tutoring system curriculum when they use handwriting interfaces *vs.* typing. Both of these results encourage future exploration of the use of handwriting interfaces for mathematic instruction online.

Keywords. Math learning, handwriting input, intelligent tutoring systems.

Introduction

Many schools throughout the United States now incorporate computers as a regular part of classroom instruction [1] and use intelligent tutoring systems (ITS) as supplements to traditional classroom instruction. Students primarily interact with these systems via keyboard-and-mouse (typing). *Output* modality (that presented to student) contrasts have been studied, including the use of animations, diagrams and talking heads (*e.g.*, [2]), but so far nothing has been reported on *input* modality (that generated by student) and learning. We believe that input modality is extraneous to the problem-solving process. This paper reports evidence in favor of handwriting-based interfaces with respect to learning in algebra equation solving. Past work has found usability benefits of handwriting for entering math on the computer [3]; our results show that handwriting input continues to have benefits when extended to a learning task.

1. Background and Motivation

While ITSes are beginning to explore natural language interfaces (*e.g.*, [4]), students still primarily interact with most systems via typing. This is partly because the technology available to most students is limited to keyboard-and-mouse. This is changing however, as students receive PDAs or TabletPCs in the classroom [1].

However, while advantages of pen-based input have been explored for the math domain in terms of usability measures [3], very little work has been done analyzing the effect of input modality on learning. One study has reported results comparing a variety of pen-based interfaces for solving geometry problems with students [5], but it does not provide a current practice (typing) control condition for comparison.

2. Experiment Design

Do students experience differences in learning due to the modality in which they generate their answers? We present two modalities in this paper: *typing*, in which students typed out the solution in a blank text box; and *handwriting*, in which students wrote the solution using a stylus in a blank space on the screen.

A total of 48 paid participants of middle and high school age participated; 50% were female. Most students had not used handwriting input on the computer before; two-thirds were very comfortable with typing. In spite of the wide range of ages, most participants were at about the same level of algebra skill based on pre-test scores. The problem-solving session used a Wizard-of-Oz design in which the students received feedback on their answer to each problem from the human experimenter. As a scaffold, students alternated copying a non-annotated worked-out example on the computer and solving an analogous equation while referring to the example (*c.f.*, [6]). We controlled for content rather than time. Dependent variables include the total time to complete all problems; time to solve each problem; number of attempts during training to either get the answer correct or move on (max=3); and change in score from pre-test to post-test.

3. Results and Discussion

Time on Task. We measured the total time students took to complete the problem-solving session. In general, typing students took about twice as long as handwriting students. We hypothesized that equations with 2D elements such as fractions would impact time only for typing; our results confirmed this. Repeated measures analysis of the average time students took *per problem* to solve problems with fractions (about half) *vs.* without fractions revealed a significant interaction between *modality* and *appearance of fractions* ($F_{2,36}=5.252$, $p<0.01$). Typing is slowed down more by the appearance of fractions, whereas there is no difference in the handwriting condition when fractions appear *vs.* when they do not. The time-speedup is not a direct measure of learning gain, but implies that students can advance farther in the curriculum when handwriting than typing.

Learning Gains and Efficiency. Despite taking about half the time, handwriting students learned just as much as typing students based on test scores. There was no significant difference between modalities in *learning gain* from pre-test to post-test ($F_{2,35}=0.293$, *n.s.*). This means that, even though students solved the same number of problems and took less time in handwriting than in typing, their learning as measured by performance improved about the same amount (mean=11.75%, stdev=17.34). This measure of learning is relatively coarse, and the standard deviation is high. In future studies we plan to analyze *how* the learning may have differed between conditions based on concept mastery. Although learning gains were of the same magnitude based

on pre- to post-test scores, the fact that the time spent per condition was so different suggests that perhaps handwriting was a more *efficient* learning modality than typing. The concept of *learning efficiency* has been used, *e.g.*, in [6], to explore how students may be able to achieve similar levels of mastery but do fewer problems. This is an area we also plan to pursue in future work.

Transfer to Paper. One advantage we hypothesize to using handwriting is that it will allow a greater degree of transfer to paper than typing. We assessed level of transfer in this study by correlating the *pre-test score* and *post-test score* with *performance during training*. We hypothesized that the cases in which there was a *modality switch* (*i.e.*, writing on the pre-test to typing in the interface to writing on the post-test) should have a lower correlation in performance during training *vs.* the tests. We ran separate bivariate correlations of percent of problems solved on the first try during training and the *pre-test score* or the *post-test score*, grouped by condition. The Pearson correlation for typing was not statistically significant for either test (0.343, $p=0.275$ for pre-test; 0.320, $p=0.310$ for post-test), whereas handwriting was significant for both (0.613, $p<0.05$ for pre-test; 0.708, $p<0.01$ for post-test). Because handwriting does not involve a modality switch from training to testing, there is a higher degree of transfer. Performance during testing more closely matches performance during training when the modality of training is similar to that of testing.

4. Conclusions and Future Work

We have reported valuable early evidence in favor of handwriting-based interfaces for ITS in math, especially algebra equation solving. Students are able to solve problems twice as quickly when handwriting than typing by virtue of increased input speed, but they seem to learn just as much as their typing counterparts based on test performance. In addition, students seem to achieve a higher degree of transfer when using handwriting on the computer than when using typing. More work is needed to establish a theory on how to achieve better learning gains using an appropriate interface.

Acknowledgments

Supported by NSF Award #SBE-03554420 and Pittsburgh Science of Learning Center. Thanks: Tom Bolster, Noboru Matsuda, Chris Atkeson, Shelley Evenson, Sue Fussell.

References

[1] Wood, C. Technology and Education. PC Magazine (2002).
[2] Mayer, R. E. Multimedia Learning. New York: Cambridge University Press, Boston, MA, 2001.
[3] Anthony, L., Yang, J., and Koedinger, K.R. Evaluation of Multimodal Input for Entering Mathematical Equations on the Computer. CHI 2005, 1184-1187.
[4] Aleven V.A.W.M.M., Koedinger, K.R., and Popescu, O. A Tutorial Dialog System to Support Self-Explanation: Evaluation and Open Questions. AIEd 2003, 39-46.
[5] Oviatt, S., Arthur, A. and Cohen, J. Quiet Interfaces that Help Students Think. UIST 2006, 191-200.
[6] Ringenberg, M. and VanLehn, K. Scaffolding Problem Solving with Annotated Worked-Out Examples to Promote Deep Learning. ITS 2006, 625-634.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Extension of IMS-QTI to Express Constraints on Template Variables in Mathematics Exercises

Odette AUZENDE, Hélène GIROIRE, Françoise LE CALVEZ
University Paris 6, LIP6, MOCAH team
104 avenue du président Kennedy, 75016 Paris

Abstract. We propose an extension of the IMS-QTI v2 specification to define templates of mathematics exercises with template variables linked by constraints. We describe the various types of constraints we add, then we present functionalities for defining template mathematics exercises and creating associated dynamic web pages

Keywords. IMS-QTI, mathematics exercise modeling, constraints, interoperability

Introduction

To train students in different domains, teachers have to author and assess numerous homework assignments, exercises, problems and tests. New technologies offer a wonderful opportunity to create exercises automatically. Our purpose is to allow teachers to define template mathematical exercises whose variables are bound by constraints in order to have numerous instances of the exercises with different values to instantiate the variables.

To allow reusability and interoperability among systems, some standards have been set: SCORM [1] and IMS-Learning Design [2] for the organisation of pedagogical contents, LOM [3] for expressing pedagogical meta-data. E-learning platforms generally use their own XML schemas to create pedagogical resources. Question and Test Interoperability (QTI) [4] defined by the IMS consortium tends to become a shared specification [5].

In the mathematical domain, some standards have been defined: MathML, OpenMath, OMDoc. These standards are used to express and share resources but are not standards for the representation of exercises and their processing.

We are working on the representation of template exercises so that teacher-authors can create them in order to generate on-line clones for remediation or for training. We started this work by using the mathematical exercise data base of Odile Jacob Multimedia (OJM) a French editor which supplies about 300 secondary schools with several hundred exercises[1]. Each exercise is designed as a template exercise associated with constraints on the variables. To enable playing of these resources on various existing e-learning platforms, we use the specification IMS-QTI v2 that we have enriched to define the constraints.

---

[1] This research was supported in part by a RIAM (network for Research and Innovation in Audiovisual and Multimedia) project including the editor OJM (Odile Jacob Multimedia).

1. IMS-QTI specification

To specify the editing and playing of exercises, IMS GLC has defined the Question and Test Interoperability specifications. Figure 1 shows the structure of the *assessmentItem* which defines an exercise in QTI v2. Two classes enable to declare and define template variables: *templateDeclaration* and *templateProcessing*.

assessmentItem

variableDeclaration	responseDeclaration	*to declare response variables*
	outcomeDeclaration	*concerns the ouput variables of the assessment process*
	templateDeclaration	*to declare template variables*
templateProcessing		*To express the rules to assign values to template variables*
itemBody		*To define the presentation of the exercise and the interaction*
responseProcessing		*To define the assessment rules and the values of the outcomes*
modalFeedback		*To define feedback to display to the user after the assessment processing*

Figure 1: Structure of an *assessmentItem* (QTI v2)

We can define variables A and B whose types are integers. Variable A is bounded by 2 and 6 and randomly valued. The value of B depends on the value of A: if A is greater than 4, B is bounded by 2 and 6, else the value of B is bounded by 10 and 15.

A and B are bounded, but **B cannot be valued before A is valued**: there is a **sequential dependence** from A to B. But, to express secondary school exercises, mathematics teachers need other template rules in order to express constraints binding the template variables when they are **interdependent**, to define a collection of template variables whose **size** is also a template variable, to express some constraints **binding the variables** contained in a template collection. We propose extensions to QTI to express such constraints.

2. IMS-QTI extensions

We enrich the class ***expression*** by adding the operators *repeat*, *sumContainer* and usual mathematical operators such as *gcd*, *lcm.* With QTI v2, it is impossible to define a template collection whose size is itself a *templateVariable*. *repeat* enables to define such a template collection: figure 2 shows the definition of a collection of N integers bounded by 1 and 20.

```
<templateDeclaration identifier="N" cardinality="single" baseType="integer" />
<templateDeclaration identifier="tab" cardinality="multiple" baseType="integer" />
....
<setTemplateValue identifier="tab" >
        <multiple>
           <repeat time="N"><randomInteger min="1" max="20" /> </repeat>
        </multiple>
 </setTemplateValue>
```

Figure 2: declaration of a template collection whose size is N

We enrich the class ***templateProcessing*** by adding the class *templateConstraint* to describe constraints. Semantics on *templateConstraint* is that the process of valuation of the variables must be reiterated until they satisfy the constraints. This extension allows us to express the constraints between A et B as:

`"A integer, B integer, B odd, A+B is multiple of 10"`

The third constraint can be expressed as "gcd(B,2)=1".The fourth constraint is expressed using the declaration of the expression S = A+B and the constraint "S is a multiple of 10" (see figure 3). In a similar manner, the constraint "the sum of all the elements of a template collection is 100" is expressed using the operator *sumContainer*:

```
<equal> <sumContainer> <variable identifier="tab" /> </sumContainer>
        <baseValue baseType="integer"> 100 </ baseValue>            </equal>
```

```
.....
<templateProcessing>
        ...
        <setTemplateValue identifier="S"> <sum> <variable identifier="A" /> <variable identifier="B" /> </sum></setTemplateValue>
        <templateConstraint>
          <not><equal><gcd><variable identifier="B" /><baseValue baseType="integer">2</baseValue></gcd>
                    <baseValue baseType="integer">1</baseValue></equal></not>
        </templateConstraint>
        <templateConstraint>
          <equal><integerModulus><variable identifier="S" /><baseValue baseType="integer">10</baseValue></integerModulus>
                       <baseValue baseType="integer">0</baseValue></equal>
        </templateConstraint>
</templateProcessing>
```

Figure 3: expression of constraints with templateConstraint

3. Functionalities for defining template mathematics exercises and creating associated dynamic web pages

We designed and implemented a "Constraint Editor" (QTI-CE). Using QTI-CE, teachers can declare variables, arithmetical expressions depending on other variables and arrays. Constraints conjunction or disjunction can be defined.

QTI-CE generates several files: the MathML file allows teachers to visualize the constraints using a browser including a MathML player; the execution of the java file gives for each new execution a set of values for the different variables satisfying the constraints; the QTI file must be incorporated into the standard *itemAssessment* file to form a complete exercise. These files are the inputs of the "Web Pages Generator" (QTI-WPG) which builds two dynamic web pages: the first one presents the exercise to the learner, waits for the responses and sends them to the second page which displays the feedback. Every time these pages are used, the values of variables are re-calculated. These web pages play the role of a "cloning engine" by directly delivering clones of the template exercise. Complete examples can be seen at: http://webia.lip6.fr/~lecalvez/QTI/

4. Conclusion

We have defined an extension of IMS-QTI v2.1 which enables teachers to create template mathematical exercises with constraints between template variables. Constraints can be expressed using operators (arithmetical, relation, logical) as in usual mathematics exercises or concern the creation of a set of template variables whose size is itself a template variable.

Using this extension, all the constraints of mathematics exercises of the OJM data base can be expressed and QTI-CE allows teacher-authors to edit them and to verify the feasibility of the instantiation of the template variables. At each running of the jsp.files generated by QTI-WPG, new clones of a template exercise are displayed to the learner.

References

[1] SCORM http://www.adlnet.org
[2] IMS-LD http://www.imsglobal.org/learningdesign/
[3] LOM http://ltsc.ieee.org/wg12
[4] IMS Question and Test Interoperability (2006), Version 2.1 public draft (revision 2) specification, available at http://www.imsglobal.org/question/ (consulted in September 2006)
[5] Vogten, H., Martens, H., Nadolski, R; Tattersall C., van Rosmalen, P., Koper, R., (2006) CopperCore Service Integration – Integrating IMS Learnng Design and IMS Question and Test Interoperability, ICALT 2006, IEEE Computer Society pp. 378-382

Do Various Self-Regulatory Processes Predict Different Hypermedia Learning Outcomes?

Roger AZEVEDO [a, c], Amy WITHERSPOON [a], Shanna BAKER [a], Jeffrey GREENE [b], Daniel MOOS [b], Jeremiah SULLINS [a], Andrew TROUSDALE [a] and Jennifer SCOTT [a]
[a]*University of Memphis, Dept. of Psychology, Institute for Intelligent Systems, 3693 Norriswood Avenue, Memphis, TN, 38152, USA*
[b]*University of Maryland, College Park, MD, 20742, USA*
[c] *razevedo@memphis.edu*

Abstract. Eighty-two (N = 82) college students with little knowledge of the circulatory system were randomly assigned either to the control condition (SRL; self-regulated learning) or human tutoring (ERL; externally-regulated learning) condition. Learners in the SRL condition regulated their own learning, while learners in the ERL condition had access to a human tutor who facilitated their self-regulated learning. All learners were given 40 minutes to learn about the circulatory system. We collected several pretest and posttest learning measures and collected think-aloud protocols during learning. Generally, the learners in the ERL condition significantly outperformed the learners in SRL condition. In comparison to the SRL condition, results indicate that learners in the ERL condition deployed significantly more SRL processes related to planning, monitoring, and handling task difficulties. Each of the classes of SRL processes was predictive of learners' performance on different posttest measures.

Introduction

Learning about complex and challenging science topics with hypermedia and other multi-representational non-linear technologies requires learners to deploy key self-regulatory processes [1]. Despite the paucity of SRL with hypermedia research, emerging research has revealed that students of all ages rarely deploy the key self-regulatory processes necessary to understand complex and challenging science topics such as the circulatory system (e.g., [2,3]). In addition, the majority of research on hypermedia learning has focused on learner's declarative knowledge and has yet to assess other types of knowledge.

The second issue is how to determine what aspects of SRL are most helpful in promoting gains in declarative knowledge and qualitative shifts in mental models necessary to truly understand complex science topics. Little research has been conducted that examines what kinds of SRL processes—cognitive, motivational/affective, behavioral, and contextual— are most helpful during the cyclical and iterative phases of planning, monitoring, control, and reflection that should take place when students learn with hypermedia environments (e.g., [1,4]). In other

words, we argue that changes from pretest to posttest on several learning measures are associated with the use of specific SRL processes and their frequency during learning. We advance that determining which SRL processes are associated with specific learning measures is an important contribution toward both helping students more effectively utilize hypermedia environments, and informing the design of adaptive hypermedia systems [5].

1. Method

Eighty-two *(N* =82) non-science college students from a public American university participated in this study. The human tutor had completed a bachelor's degree in biology and was currently enrolled in a PhD program in Educational Psychology. We used the matching and labeling tasks, mental model essay, and blood flow diagram pretest and posttest measures developed and used by Azevedo and colleagues [see 1, 2, 3 for details].

During the experimental phase, the learners used a hypermedia environment to learn about the circulatory system. Learners were allowed to use all of the system features including the search functions, hyperlinks, table of contents, and multiple representations of information. They were also allowed to navigate freely within the environment. They were all shown the content (e.g., circulatory system) which contained multiple representations of information (i.e., text, diagrams). Participants were randomly assigned to conditions: SRL Condition (n = 41), and ERL Condition (n = 41). The students in the ERL condition were tutored by a human tutor who followed a tutoring script that prompted students to deploy certain self-regulatory processes during learning. Details regarding the script, the coding of the matching, labeling, blood flow, and mental model tasks are described in [3].

2. Results

Question 1: Do students in different tutoring conditions deploy significantly different numbers of self-regulatory processes during learning with hypermedia?

We conducted several independent t-test analyses to determine whether participants deployed statistically significantly different number of SRL moves during the 40-minute learning task. Overall, learners in the ERL condition deployed statistically significantly more self-regulatory moves during the learning task (M_{SRL}= 65.85 vs. M_{ERL}= 92.95; see Table 1). More specifically, they used 1.6 times more self-regulatory processes than those in the SRL condition. This translates to an average of 2.3 self-regulatory moves a minute. In addition, we also found statistically significant results between the conditions for three out of the five classes of SRL processes. More specifically, learners in the ERL condition used significantly more planning,

Table 1. *Means and (Standard Deviations) for Total Coded Raw SRL Frequencies across Tutoring Conditions*

Class of SRL Processes	SRL Condition (n = 41)	ERL Condition (n = 41)
Planning*	5.90 (5.56)	16.24 (11.73)
Monitoring*	14.73 (14.71)	29.34 (14.86)
Learning Strategies	34.54 (18.67)	32.22 (14.79)
Task Difficulties and Demands*	8.85 (5.23)	13.44 (8.02)
Interest	1.41 (2.06)	1.45 (1.83)
*Total SRL Processes**	*65.85 (33.41)*	*92.95 (36.11)*

Note: * $p < .05$

monitoring, and handling task difficulties and demands (than learners in the SRL condition).

Question 2: Can we predict learners' posttest performance scores based on their use of self-regulatory processes during learning with hypermedia?

We ran several multiple regression analyses to determine whether certain classes of SRL processes (e.g., planning) were predictive of learners' performance on the four posttest measures. In each stepwise regression analysis we entered learners' total planning, monitoring, learning strategies, and task difficulties and demands as predictor variables and the corresponding posttest scores as the dependent variable.

Table 2. *Summary of Multiple Regression Analyses for SRL Processes Predicting Dependent Measures*

	Final Model		
	b	SE (*b*)	β
Labeling Posttest			
Planning	1.240	.249	.494[a]
Flow Diagram Posttest Task Difficulty and Demands	1.923	.550	.370[b]
Mental Model Posttest			
Monitoring	.051	.018	.307[c]

Note. R^2 = .244 for Labeling Posttest; R^2 = .137 for Flow Diagram Posttest; R^2 = .094 for Mental Model Posttest; [a] $p<.001$; [b] $p = .001$; [c] $p = .006$

A stepwise multiple regression analysis was run to determine which class of SRL processes were significant predictors of the labeling posttest scores. It revealed that *Planning* was a statistically significant predictor for the labeling posttest dependent measure, $F\,(1, 77) = 24.868$, $p < .001$; $t\,(77) = 4.987$, $p < .001$. Those participants who used planning more often scored higher on the labeling posttest measure. Another stepwise multiple regression analysis was run to determine which SRL processes were significant predictors of the flow diagram posttest scores. It revealed that the SRL processes related to handling *Task Difficulty and Demands* was a statistically significant predictor for the flow diagram posttest dependent measure, $F\,(1, 77) = 12.219$, $p = .001$; $t\,(77) = 3.496$, $p = .001$. Those participants who used processes related to handling task difficulties and demands more often scored higher on the flow diagram posttest measure. A stepwise multiple regression analysis was also run to determine which SRL processes were significant predictors of the mental model posttest scores. It revealed that *Monitoring* was statistically significant predictor for the mental model posttest measure, $F\,(1, 77) = 8.033$, $p = .006$; t (77) = 2.834, $p = .006$. Those participants who monitored several aspects of their learning more often scored higher on the mental model posttest measure. No SRL process variables were found to be significant predictors for the matching posttest scores.

References

[1] Azevedo, R. (2005). Using hypermedia as a metacognitive tool for enhancing student learning? The role of self-regulated learning. *Educational Psychologist, 40(4), 199-209*.

[2] Azevedo, R., & Cromley, J.G. (2004). Does training on self-regulated learning facilitate students' learning with hypermedia? *Journal of Educational Psychology, 96*(3), 523-535.

[3] Azevedo, R., Moos, D.C., Greene, J.A., Winters, F.I., & Cromley, J.C. (in press). Why is externally-regulated learning more effective than self-regulated learning with hypermedia? *ETR&D*

[4] Shapiro, A., & Niederhauser, D. (2004). Learning from hypertext: Research issues and findings. In D. Jonassen (Ed.). *Handbook of Research for Ed. Communications and Tech. (2nd ed).* Mahwah, NJ: Erlbaum.

[5] Lajoie, S., & Azevedo, R. (2006). Teaching and learning in technology-rich environments. In P. Alexander & P. Winne (Eds.), *Handbook of educational psychology* (2nd ed.) (pp. 803-821). Mahwah, NJ: Erlbaum.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Assessing Learning in a Peer-Driven Tutoring System

Ari Bader-Natal *, Jordan Pollack
DEMO Lab, Brandeis University

Abstract.
In many intelligent tutoring systems, a detailed model of the task domain is constructed and used to provide students with assistance and direction. Reciprocal tutoring systems, however, can be constructed without needing to codify a full-blown model for each new domain. This provides various advantages: these systems can be developed rapidly and can be applied to complex domains for which detailed models are not yet known. In systems built on the reciprocal tutoring model, detailed validation is needed to ensure that learning indeed occurs. Here, we provide such validation for SpellBEE, a reciprocal tutoring system for the complex task domain of American-English spelling. Using a granular definition of response accuracy, we present a statistical study designed to assess and characterize student learning from collected data. We find that students using this reciprocal tutoring system exhibit learning at the word, syllable, and grapheme levels of task granularity.

American-English spelling is a domain known to be difficult to fully codify [5]. It serves as the task domain addressed by the SpellBEE reciprocal tutoring system, which is the first of a growing suite built on the BEEweb model [2]. SpellBEE has been publicly available online (at SpellBEE.org) for the past three years and has, to date, collected data from over 17,000 active participants engaged in over 22,000 completed peer-tutoring sessions. Participants primarily consist of American students in grades 2 through 8.

Student interactions in the BEEweb model are strictly governed by a two-student reciprocal tutoring arrangement, similar to prior classroom-based protocols compiled by O'Donnell and King [6] and the computer-based protocols of Chan and Chou [3]. This reciprocal tutoring protocol determines the structured flow of interactions between a pair of students. A game theoretic framework, the "Teacher's Dilemma," serves as a motivational mechanism within the game. Each step in the reciprocal tutoring protocol corresponds with some aspect of this game. Details on the BEEweb model and the Teacher's Dilemma can be found in our earlier work [1,2], so the protocol is only briefly described here: Once a pair of students has elected to engage in a collaboration session, they proceed through a sequence of steps (repeated once during each turn of the game), alternating between playing the roles of "tutor" and "tutee." At the first step, a student is asked to construct a challenge to pose to their partner (In SpellBEE, this construction task takes the form of choosing from a short list of words randomly selected from a 3000+ word dictionary.) In the second step, the student attempts to solve the challenge that was posed by their partner in the previous step (In SpellBEE, the student sees a sentence context

*Correspondence to: Ari Bader-Natal, Brandeis University, Computer Science Department MS018. Waltham, MA 02454. USA. Tel.: +1 781 736 3366; Fax: +1 781 736 2741; E-mail: ari@cs.brandeis.edu.

for the word with the challenge word hidden, hears the entire sentence read aloud by text-to-speech software, and then types a response spelling into a text field.) When the student submits this response, they receive feedback on its accuracy and are shown the correct response if they made a mistake. Finally, the student is presented with feedback on how their partner fared on the problem that they selected in the first step, which can be useful when constructing future challenges.

Given the peer-driven nature of this design, we need to validate that students using SpellBEE are indeed learning and improving at spelling. To do this, we first define a fine-grained concept of response accuracy and select two domain-appropriate notions of sub-problem structure to examine, *syllables* and *graphemes*.[1] We then run statistical tests to identify if and how SpellBEE students improve at spelling, both overall and with respect to these two sub-word structures.

Past analysis of SpellBEE has been based on a scalar measure of challenge difficulty and a dichotomous measure of response accuracy [2]. We refer to this accuracy measure here as *whole-word accuracy*, which is defined in terms of a student's response (spelling), r, to the (word) challenge posed to them, c:

$$\mathcal{A}(c,r) = \begin{cases} 0 & \text{if response } r \text{ is } \textit{not} \text{ a correct solution to challenge } c \\ 1 & \text{if response } r \text{ is a correct solution to challenge } c \end{cases}$$

In order to gain a deeper understanding of the nature of student improvements within the spelling domain, we need to analyze spelling accuracy at a finer grain. Hanna et al. [5] thoroughly detailed the role and regularity of phoneme-grapheme correspondences in American-English spelling, and we draw upon in this work by examining spelling accuracy at the levels of graphemes and syllables. We introduce a concept of *sub-word accuracy*, defined with respect to a sub-word structure s, such as a grapheme or syllable:

$$\mathcal{A}'(c,r,s) = \begin{cases} 0 & \text{if sub-problem } s \text{ of challenge } c \text{ was } \textit{not} \text{ correctly solved in } r \\ 1 & \text{if sub-problem } s \text{ of challenge } c \text{ was correctly solved in } r \end{cases}$$

These two definitions of accuracy can be leveraged to construct statistical tests for learning. As SpellBEE does not currently incorporate pre- and post-testing, we rely on McNemar's test to examine the effect of SpellBEE usage on spelling improvement. This is a non-parametric statistical method that tests for change in a dichotomous trait within a group of subjects before and after an intervention [4]. When a student sees some challenge c at time t_i and later at t_j, the change from $\mathcal{A}(c,r_i)$ to $\mathcal{A}(c,r_j)$ can indicate learning. For some fixed challenge c, let Δ be the number of students for whom $\mathcal{A}(c,r_i) < \mathcal{A}(c,r_j)$, and let ∇ be the number of students for whom $\mathcal{A}(c,r_i) > \mathcal{A}(c,r_j)$.[2] McNemar's test uses Δ and ∇ to test the association between SpellBEE usage and response accuracy[3] (using the statistic: $\chi^2_{McNemar} = \frac{|\Delta-\nabla|^2}{\Delta+\nabla}$.) With this approach, we find that the association between whole-word spelling accuracy and SpellBEE usage is significant, with accuracy improving with usage (Yates' continuity-corrected $\chi^2 = 28.2031$, $df = 1, p < 0.001$, odds ratio $= 1.9891$).

Based on the finer-grained $\mathcal{A}'$ accuracy, we can use McNemar's test to see if students are learning the spelling of sub-word structures that occur in many different words.

[1] A *phoneme* is the smallest unit of sound in a language, and a *grapheme* is the written form of the phoneme.
[2] If more than one comparison is available for a student, the one with the most elapsed time $(t_j - t_i)$ is used.
[3] Note that the number of students for whom $\mathcal{A}(c,r_i) = \mathcal{A}(c,r_j)$ is not used.

Table 1. Graphemes are classified according to how student spelling accuracy changed during SpellBEE usage.

Improved ($p < 0.05$)	No Significant Change ($p \geq 0.05$)	Worsened ($p < 0.05$)
A, C, CC, CE, CQ, CQU, CT, D, DG, E, ED, EI-E, EIGH, EN, ES, F, G, GH, GI, H, I, I-E, IA, IA-E, IGH, IN, J, K, KN, L, M, N, NG, O, OL, ON, OO, OW, OW-E, P, PP, PT, Q, QU, R, S, SI, SSI, ST, T, TH, TI, U, U-E, W, WH, X, Y	A-E, AI, AI-E, AL, AU, AW, AY, B, CH, CI, CK, DD, DI, E-E, EA, EA-E, EE, EE-E, EI, EL, EO, ET, EW, EY, –EY, FF, FT, GG, GN, GU, GUE, IE, IE-E, IL, LD, LE, LL, LV, MB, MM, MN, NN, O-E, OA, OI, OU, OUGH, OWE, RR, SC, SCI, SH, SL, SS, SW, TCH, TT, UE, UI, UI-E, V, WR, Z	*none*

For words c_x and c_y containing a common *grapheme* s, we now let Δ count the students for whom $\mathcal{A}'(c_x, r_i, s) < \mathcal{A}'(c_y, r_j, s)$ and let ∇ count the students for whom $\mathcal{A}'(c_x, r_i, s) > \mathcal{A}'(c_y, r_j, s)$. Table 1 shows that at the $\alpha = 0.05$ level, we found 58 graphemes for which student spelling accuracy significantly changed for the better, 63 graphemes for which no significant change was observed, 0 graphemes for which student spelling accuracy significantly changed for the worse, and 50 graphemes for which not enough data was available to use the test (i.e. $\Delta + \nabla < 10$). Similarly, when we examine changes in response accuracy over time at the granularity of *syllables* as sub-structure s (for which enough data was available to use McNemar's method), we find that students significantly improved on 79 syllables, exhibited no significant change on 304 syllables, and significantly worsened on 0 syllables.

Notably, we observed no syllable or grapheme for which student spelling significantly worsened after SpellBEE usage, and many for which student spelling significantly improved. Results from all three levels of analysis granularity support the conclusion that students are improving at the spelling task with usage of the tutoring system, and the analysis allows us see how this progress is distributed across sub-problem structures. Finally, we find that sub-word accuracy can be used as the basis for a principled statistical validation of learning in the SpellBEE reciprocal tutoring system. By choosing domain-appropriate definitions of sub-problem accuracy, the methodology used here can be applied to analyzing student learning other tutoring systems.

References

[1] Ari Bader-Natal and Jordan Pollack. Motivating appropriate challenges in a reciprocal tutoring system. In C.-K. Looi, G. McCalla, B. Bredeweg, and J. Breuker, editors, *Proc. of the 12th Intl. Conf. on AI in Education (AIED-2005)*, pages 49–56, Amsterdam, July 2005. IOS Press.

[2] Ari Bader-Natal and Jordan Pollack. BEEweb: A multi-domain platform for reciprocal peer-driven tutoring systems. In M. Ikeda, K. Ashley, and T.-W. Chan, editors, *Proc. of the 8th Intl. Conf. on Intelligent Tutoring Systems (ITS-2006)*, pages 698–700. Springer, June 2006.

[3] Tak-Wai Chan and Chih-Yueh Chou. Exploring the design of computer supports for reciprocal tutoring. *International Journal of Artificial Intelligence in Education*, 8:1–29, 1997.

[4] B. S. Everitt. *The Analysis of Contingency Tables*. Chapman and Hall, 1977.

[5] Paul Hanna, Jean Hanna, Richard Hodges, and Edwin Rudorf. Phoneme-grapheme correspondences as cues to spelling improvement. Report OE-32008, Office of Education, 1966.

[6] Angela M. O'Donnell and Alison King, editors. *Cognitive Perspectives on Peer Learning*. Lawrence Erlbaum Associates, 1999.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Evaluating a Collaborative Constraint-based Tutor for UML Class Diagrams

Nilufar BAGHAEI and Antonija MITROVIC
Department of Computer Science and Software Engineering
University of Canterbury, Private Bag 4800, Christchurch, New Zealand
nilufar.baghaei@gmail.com, tanja@cosc.canterbury.ac.nz

Abstract. COLLECT-UML is a collaborative constraint-based tutor for teaching object-oriented analysis and design using Unified Modelling Language. It is the first system in the family of constraint-based tutors to represent a higher-level skill such as collaboration using constraints. We present the full evaluation study carried out at the University of Canterbury to assess the effectiveness of the system in teaching UML class diagrams and good collaboration. The results show that COLLECT-UML is an effective educational tool. In addition to improved problem-solving skills, the participants both acquired declarative knowledge about good collaboration and did collaborate more effectively. The participants have enjoyed working with the system and found it a valuable asset to their learning.

1 COLLECT-UML

COLLECT-UML [1, 2] is a web-based collaborative constraint-based tutor, teaching Object-Oriented (OO) analysis and design using Unified Modelling Language (UML). Constraint-based tutors have been successfully used in the past to support individual learning in a variety of domains. COLLECT-UML is the first constraint-based tutor supporting collaborative learning. This paper briefly describes the system and presents a full evaluation study conducted at the University of Canterbury to examine the system's effectiveness in teaching UML and successful collaboration. COLLECT-UML provides feedback on both collaboration issues (using the collaboration model, represented as a set of meta-constraints) and task-oriented issues (using the domain model, represented as a set of syntax and semantic constraints).

We started by developing a single-user version. The system was evaluated in a real classroom, and the results showed that students' performance increased significantly. For details on the architecture, interface and the results of the evaluation studies conducted using the single-user version, refer to [2].

The student interface is shown in Figure 1. The problem description pane presents a design problem that needs to be modelled by a UML class diagram. Students construct their individual solutions in the private workspace (right). They use the shared workspace (left) to collaboratively construct UML diagrams while communicating via the chat window (bottom). The system provides feedback on the individual solutions, as well as on group solutions and collaboration. The domain-level feedback on both individual and group solutions is offered at four levels: *Simple Feedback*, *Error flag*, *Hint* and *All Hints*. The collaboration-based advice is given to individual students based on the content of the chat area, the student's contributions to

the shared diagram and the differences between student's individual solution and the group solution. For more details on the interface, refer to [1].

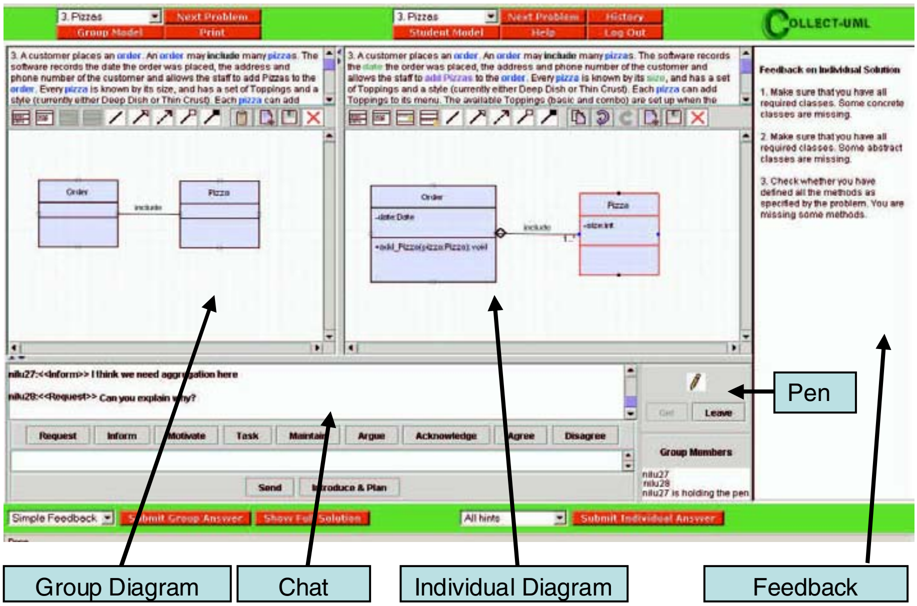


Figure 1. COLLECT-$\mathcal{UML}$ interface

The goal of our research is to support collaboration by modelling collaborative skills. COLLECT-$\mathcal{UML}$ is capable of diagnosing students' collaborative actions, such as contributions to the chat area and contributions to the group diagram, using an explicit model of collaboration. This collaboration model is represented using constraints, the same formalism used to represent domain knowledge. A significant contribution of our work is to show that constraint can be used not only to represent domain-level knowledge, but also higher-order skills such as collaboration. For more details about the meta-constraints refer to [1].

2 Evaluation

We conducted an evaluation study at the University of Canterbury in May 2006. The study involved 48 volunteers enrolled in an introductory Software Engineering course. The students learnt UML modelling concepts during two weeks of lectures and had some practice during two weeks of tutorials prior to the study. The study was carried out in two streams of two-hour laboratory sessions over two weeks. In the first week, the students filled out a pre-test and interacted with the single-user version of the system. Doing so gave them a chance to learn the interface and provided us with an opportunity to decide on the pairs and moderators.

At the beginning of the sessions in the second week, we told students what characteristics we would be looking for in effective collaboration (that was considered as a short training session). The instructions describing the characteristics of good collaboration and the process we expected them to follow were also handed out. The students were randomly divided into pairs with a pre-specified moderator. The moderator for each pair was the student who had scored higher in the pre-test.

The experimental group consisted of 26 students (13 pairs) who received feedback on their solution as well as their collaborative activities. The control group consisted of 22 students (11 pairs) who only received feedback on their solutions (no feedback on collaboration was provided in this case). All pairs received instructions on characteristics of good collaboration at the beginning of the second week. The total time spent interacting with the system was 1.4 hours for the control and 1.3 hours for the experimental group.

There was no significant difference on the pre-test results, meaning that the groups were comparable. The students' performance on the post-test was significantly better for both control group ($t = 2.11$, $p = 0.01$) and experimental group ($t = 2.06$, $p = 0.002$). The experimental group, who received feedback on their collaboration performed significantly better on the collaboration question ($t = 2.02$, $p = 0.003$), showing that they acquired more knowledge on effective collaboration. The effect size on student's collaboration knowledge is also very high: 1.3. The experimental group students contributed more to the group diagram, with the difference between the average number of individual contribution for control and experimental group being statistically significant ($t = 2.03$, $p = 0.03$).

Figure 2 illustrates the probability of violating a meta-constraint plotted against the occasion number for which it was relevant, averaged over all meta-constraints and all participants in the experimental group. There is a regular decrease, thus showing that students learn meta-constraints over time. Because the students used the system for a short time only, more data is needed to analyze learning of meta-constraints, but the trend identified in this study is encouraging.

The results of both subjective and objective analysis proved that COLLECT-*UML* is an effective educational tool. The experimental group students acquired more declarative knowledge on effective collaboration, as shown by their higher scores on the collaboration test. The collaboration skills of the experimental group students were better, as evidenced by these students being more active in collaboration, and contributing more to the group diagram. All students improved their problem-solving skills as they performed significantly better on the post-test. Finally, the students enjoyed working with the system and found it a valuable asset to their learning (as specified in the questionnaires filled out at the end of the session). The results, therefore, show that constraint-based modelling is an effective technique for modelling and supporting collaboration in CSCL environments.

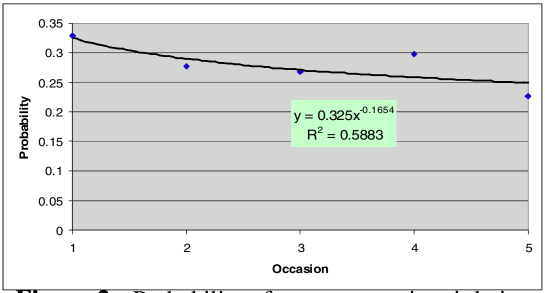


Figure 2. Probability of meta-constraint violation

References

[1] Baghaei, N., Mitrovic, A.: A Constraint-based Collaborative Environment for Learning UML Class Diagrams. ITS 2006, (2006) 176-186

[2] Baghaei, N., Mitrovic, A., Irwin, W.: Problem-Solving Support in a Constraint-based Tutor for UML Class Diagrams, Technology, Instruction, Cognition and Learning Journal, 4(1-2) (2006) (in print)

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Intelligent Tutoring and Human Tutoring in Small Groups: An Empirical Comparison

Carole R. BEAL, Erin SHAW and Mike BIRCH
Information Sciences Institute - USC
4676 Admiralty Way, Marina del Rey CA 90292 USA
{ cbeal, shaw, mbirch }@isi.edu

Abstract. The efficacy of a tutoring system for pre-algebra instruction plus human tutoring was compared to instruction provided to small groups of middle school students by experienced human math tutors, with instructional time held constant. Students completed pre- and post-tests of computation, fractions, algebra and rational numbers skills. Results indicated that students showed significant improvement from pre- to post-test, but there was no difference as a function of type of tutoring. The findings help to establish the efficacy of ITS instruction relative to skilled human tutoring of students in small groups.

Introduction

Individualized instruction is generally recognized as the “gold standard” for instruction. Students learn most rapidly and deeply when they can work one-on-one with an experienced human tutor. Research indicates that the instruction provided by intelligent tutoring systems is also effective [1, 2]. However, to date tutoring systems have typically been compared to whole-class instruction, i.e., “business-as-usual” instruction. We do not yet know how tutoring systems compare to human instruction provided in a more individualized manner. Ideally, this would involve a comparison of an ITS to one-on-one instruction with a human tutor, yet this situation is difficult to arrange in the context of in-vivo classroom research. However, as an initial investigation, in the present study we compared the efficacy of an ITS to instruction provided by experienced human tutors to small groups of students.

1. Method

1.1 Data sources

Participants included 32 middle school students (incoming Grade 7) attending an academic summer program held at the university campus. Students attended the program every Saturday for four hours, with instructional time divided between language arts (2 hours) and mathematics (2 hours). The program was structured so that students worked in small groups, with 6-8 students per adult tutor. Tutors for the math sessions were experienced mathematics teachers from local schools.

1.2 Design

For the present study, two groups of students for each of two tutors were assigned either to receive one hour of computer tutoring and one hour of human tutoring for each math session, or to receive two hours of human tutoring (small-group instruction) per session, for 6 sessions. Human tutors provided blackboard lessons and worksheet practice.

1.3 Computer tutoring

Students worked with AnimalWatch, a web-based tutoring system in which instruction in pre-algebra mathematics was integrated with environmental science content. The program presented word problems about endangered species. Problem selection and math topic were determined by the pedagogical model, based on the student's ongoing problem solving performance. Students also practiced basic math skills through AnimalWatch SkillBuilder units on math facts, decimal equivalents, rounding and estimating, and other pre-algebra material.

AnimalWatch provided immediate feedback when an answer was submitted. Incorrect answers elicited a text message, e.g., "Your work is improving but that's not correct". A second incorrect answer elicited an operations hint, e.g., "Are you sure you are adding 312 and 434?". Students who still needed help after the text hints could click a "Hints" icon to view complete worked examples, as well as interactive hints. Students' answers and use of hints were logged automatically into the database.

1.4 Assessments and scoring

All students completed a pre-test before the tutoring sessions, followed by a post-test at the conclusion of the program. The tests included 30 items: ten computation items; 6 fractions items; 6 algebra items; and 8 items assessing rational numbers skills such as ratios, proportions, and unit conversions. Tests were constructed so that items had multiple clones that were equivalent in difficulty. Students completed the tests in 45 minute sessions under the supervision of the program instructors. Students received scores for the proportion of computation, fractions, algebra and rational number items completed correctly on the pre- and post-tests.

2. Results

2.1 Pre-test

An ANOVA with Tutoring Condition (Small Group; ITS + Small Group) as the grouping factor and pre-test proportion correct scores (Computation, Fractions. Algebra, Rational Numbers) as dependent measures. Results indicated no differences in performance as a function of Tutoring Condition, indicating that students in the two tutoring conditions had similar skills before the program activities began. There was a significant effect of Math Topic: Students scored highest on Computation (65% correct) and Fractions (66% correct) items; then Algebra (52%), with the lowest scores for Rational Numbers (13% correct). Mean scores are shown in Table 1.

2.2 Pre- to post-test comparisons

To evaluate the impact of the program, an ANOVA was conducted with Tutoring Condition as the grouping factor, Test (Pre-, Post-) as a within-subjects factor, and total proportion correct on each test as outcome measures. There was a significant effect of Test, $F(1,23) = 27.82$, $p < .001$. As may be seen in Table 1, students showed significant improvement from pre- to post-test. However, there was no effect of Tutoring Condition, indicating that both groups improved to the same extent.

Table 1. Mean proportion correct for Pre- and Post-tests (Standard deviations in parentheses)

Group:	Computation	Fractions	Algebra	Rational Nos.
Only human tutoring Pre test	0.62 (0.19)	0.62 (0.36)	0.47 (0.45)	0.09 (0.17)
ITS + Human tutoring Pre test	0.66 (0.19)	0.63 (0.37)	0.49 (0.40)	0.13 (0.28)
Only human tutoring Post test	0.74 (0.13)	0.74 (0.20)	0.80 (0.19)	0.49 (0.27)
ITS + Human tutoring Post test	0.72 (0.19)	0.73 (0.24)	0.84 (0.20)	0.45 (0.40)

To investigate the math topic where tutoring had most impact, we conducted a repeated measure ANOVA comparing Pre and Post test scores of each of the four math topics, with Group as a between subjects factor. There were no effects associated with the type of tutoring (only human tutoring, or half-human and half-ITS tutoring). For Computation and Fractions problems, the increase from pre- to post-test, although consistent in direction and across participants, was not significant. For Algebra items, there was a significant improvement from pre- to post-test, $F(1,26) = 63.77$, $p < .001$. There was also a significant improvement for items involving rational numbers concepts, $F(1,25) = 25.82$, $p < .001$. This suggests that tutoring had most impact on the most challenging material.

3. Discussion

The results of the study suggest that ITS instruction can be a valid supplement for instruction. When instructional time was held constant, students who received half of their classroom learning opportunity at the computer performed as well on the assessments as students who received all instruction from an experienced human tutor working with small groups. Both program formats were effective: students showed significant increases in math skill from pre- to post- test, particularly on the more difficult topics such as fractions, ratios, proportions, and equivalents.

The next step is to drill more deeply in to the patterns of ITS usage exhibited by individual students who received ITS instruction for half of their instructional time. For example, we would expect that a student who showed significant improvement on fractions problems on the assessments should have received ITS problems and scaffolding in that topic. Failing to find such a relation would suggest an alternative interpretation focusing more on the motivational benefits of computer-based practice.

Acknowledgments

The study was supported by a grant from the Institute of Education Sciences Mathematics and Science Education program. We would like to thank the program tutors and program director Kim Thomas-Barrios for their assistance with the study, as well as project members Lei Qu, Jean-Philippe Steinmetz, Hyokyeong Lee, Sharan Volin, and Ivon Arroyo.

References

[1] Beal, C. R., Walles, R., Arroyo, I., & Woolf, B. P. (2007). On-line tutoring for math achievement testing: A controlled evaluation. Journal of Interactive Online Learning, 6, 43-55.

[2] Koedinger, K. R., Corbett, A. T., Ritter, S., & Shapiro, L. J. (2000). Carnegie Learning's Cognitive Tutor: Summary of research results. White Paper. Pittsburgh PA: Carnegie Learning.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Student Motivation and Performance in Scientific Problem Solving Simulations

Carole R. BEAL and Ronald H. STEVENS
Information Sciences Institute – USC
4676 Admiralty Way, Marina del Rey CA 90292
cbeal@isi.edu, immexr@gmail.com

Abstract. Students do not always use efficient strategies to solve scientific problems. Students' motivation beliefs were assessed through an on-line survey instrument, along with performance on easy and challenging multimedia science problem sets. Results indicated that female students reported lower self-efficacy and higher concerns about their performance than male students. Students' initial strategies predicted overall performance on the harder problem sets. In addition, although there was no difference in initial strategy, students' motivational beliefs predicted overall performance. Results suggest that both cognitive and motivational factors influence students' strategic efficacy.

Introduction

Scientific problem solving is now recognized as an important component of science education, yet recent assessments indicate that American students do not perform well on problem solving tasks [1, 2, 3]. One reason may be that traditional science instruction often fails to integrate scientific content with experiences that provide opportunities for students to develop skills of problem solving. In spite of recent interest in inquiry-based learning, students in the United States still have much more opportunity to learn *about* science than to *do* science [2].

Multimedia simulations can provide students with important opportunities to practice problem solving skills such as identifying relevant information, developing hypotheses, and tracking progress to the solution. IMMEX is a web-based multimedia simulation environment that delivers authentic science problems to students, aligned with academic content standards. For example, in the COINS problem, the student reads a prologue in which he or she is given a handful of strange coins and must identify the metal as part of a job interview at a chemicals firm. Students can request data and order various tests as they develop their hypothesis about the answer, e.g., using a virtual scale to gain information about the coins' mass. There is a cost associated with each request for information. When they submit an answer, they receive immediate feedback. Each problem includes multiple cases with the same general framework, but different specifics. For example, the COINS cases all involve the same goal of identifying a metallic substance, but the type of metal varies from case to case. Thus, the specific set of information needed to solve the problem is unique to each case, and the cases vary in difficulty.

As students solve IMMEX problems, their actions are logged to provide an integrated assessment about their strategic approach, including what information was viewed, what tests were ordered and in what sequence, and the solution accuracy. Previous work using Hidden Markov Models indicates that many students adopt inefficient strategies, such as ordering all available tests or guessing [4]. Students' initial strategies can persist over multiple cases and problem sets, sometimes for up to four months [5]. A student's overall proficiency is indicated by his or her IRT score.

The goal of the present project was to learn if student motivation might contribute to strategy selection, i.e., students with less confidence in their ability might adopt more cautious strategies, and resist modifying their approach even when the approach was not always successful.

1. Factors influencing students' problem solving strategies

1.1 Assessment of science motivation

We developed the Science Motivation Profile: an on-line instrument based on previous work with paper-and-pencil and on-line assessments [6, 7]. The instrument included items to assess five constructs identified in the motivation literature: SELF EFFICACY IN SCIENCE, EXPECTED SUCCESS IN SCIENCE, PERCEIVED DIFFICULTY OF SCIENCE, LIKING OF SCIENCE, and BELIEF THAT SCIENCE IS IMPORTANT TO LEARN. Students clicked on a Likert-type rating scale to respond. Ratings were averaged across items for each construct, yielding five scores for each student.

1.2 Data sources

The Science Motivation Profile was completed by 406 students before they worked with IMMEX. Students completed cases in COINS and a second more difficult problem set, PERIODIC TRENDS. Initial strategies and overall IRT scores served as performance measures.

2. Results and Discussion

2.1 Science motivation groups

K-means clustering conducted on the Science Motivation Profile scores suggested that there were four groups of students, informally labeled as follows:

- "confident high achievers" (N = 123): These students had the highest self efficacy beliefs, were most likely to view science classes as easy, and liked science more than other students.
- "nervous high achievers" (N = 126): Students in this group placed high value on science and liked it, but had relatively low self efficacy scores.
- "discouraged students" (N = 58): These students had the lowest scores for self-efficacy, expected success and liking of science, and were least likely to view it as valuable.
- "not-interested" (N = 99): These students thought they would do well but did not particularly like science or think it was important to learn.

Table 1. Mean scores for responses on Science Motivation Profile for Clusters

Cluster:	Like science	Ability	Value	Difficulty	Success
"Discouraged"	1.93	2.39	2.49	1.85	2.27
"Nervous"	3.85	4.13	4.30	3.25	4.02
"Not interested"	2.69	3.55	3.23	2.73	3.47
"Confident"	3.36	3.09	3.94	2.06	2.94

A cross-tabulation of gender and Motivation Cluster indicated a significant relation, χ^2 (3,400) = 25.267, $p < .001$. Female students were more likely than males to be Nervous High Achievers; in contrast, Discouraged students were more likely to be male.

2.2 Motivation and problem solving

To evaluate the contribution of initial strategy and motivation to overall performance, a MANOVA was conducted with students' IRT scores (COINS, PERIODIC TRENDS) as the within-subjects factor. Grouping factors were student's initial strategies for COINS and TRENDS, and Motivation Cluster. The results indicated that, consistent with prior work, students performed better on the COINS problem than on TRENDS, $F(1,328) = 232.61$, $p < .001$. Initial problem solving strategy did not predict overall IRT score for COINS, but did predict overall performance for TRENDS, $F(4,328) = 4.10$, $p < .01$. This indicates that as problems become harder, it is more important for students to adopt an effective strategy early on, in terms of how much they learn. In addition, there was a significant contribution of Motivation Cluster, $F(3,328) = 3.79$, $p < .01$. This indicates that students' beliefs about their performance also contribute to their problem solving performance, as indicated by their overall IRT score. Mean scores may be viewed in Table 2. Post-hoc contrasts (alpha set to 0.05) indicated that students with lower self confidence (Discouraged and Nervous students) had lower IRT scores when compared to students with higher self confidence in their ability (Not-interested Achievers and Confident Achievers). There were no significant interactions.

The results indicate that students' overall performance on problem solving tasks is influenced by their initial strategic approach, and also by the constellation of beliefs about their own performance, and the importance of doing well in science.

Table 2. Mean IRT score for COINS and TRENDS problem sets for four Motivation Clusters

Cluster:	COINS	TRENDS
"Discouraged"	58.0	40.3
"Nervous"	59.0	42.8
"Not-interested"	61.2	43.5
"Confident"	62.0	43.2

Acknowledgments

Support for the project is provided by a grant from the Institute of Education Sciences Cognition and Student Learning program. The views expressed are not necessarily those of the funding agency.

References

[1] Augustine, N. R. (2005). National Academies Committee on Prospering in the Global Economy of the 21st Century, Rising Above The Gathering Storm: Energizing and Employing America for a Brighter Economic Future. National Academies Press, Washington, D.C.
[2] National Research Council. (2006). Washington DC.
[3] OECD (2004). Problem Solving for Tomorrow's World First Measures of Cross-Curricular Competencies from PISA 2003. C. Shewbridge & A. Schleicher (Ed.s), Programme for International Student Assessment, Organization for Economic Co-operation and Development (OECD). Paris, France.
[4] Stevens, R., Johnson, D. F., & Soller, A. (2005). Probabilities and predictions: Modeling the development of scientific competence. Cell Biology Education, 4, 47-53.
[5] Stevens, R., Soller, A., Cooper, M., & Sprang, M. (2004). Modeling the development of problem solving skills in chemistry with a web-based tutor. In J. C. Lester et al. (Eds.), Intelligent tutoring systems, 7th International Conference Proceedings. Berlin: Springer-Verlag.
[6] Boekaerts, M. (2002). The on-line motivation questionnaire. In P. Pintrich & M. L. Maher (Eds.), Advances in motivation and achievement, Vol. 12: New directions in measures and methods (pp. 77-120). New York: JAI.
[7] Eccles, J., Wigfield, A., Harold, R. D., & Blumenfeld, P. (1993). Age and gender differences in children's self and task perceptions during elementary school. Child Development, 64, 830-847.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Developing a Text Support Tool for English Language Learners

Jill BURSTEIN, Jane SHORE, John SABATINI, Yong-Won LEE and Matthew VENTURA
Educational Testing Service, Princeton, New Jersey
{jburstein, jshore, jsabatini , ylee, mventura}@ets.org

Abstract. K-12 teachers often need to develop text adaptations of authentic classroom texts as a way to provide reading-level appropriate materials to English language learners in their classrooms. Adaptations are done by hand, and include practices such as, writing text summaries, substituting appropriate vocabulary, and offering native language support. In light of the labor-intensiveness involved in the manual creation of text adaptations, we have implemented the Automated Text Adaptation Tool (ATA v.1.0). This tool uses natural language processing (NLP) capabilities to automatically generate text adaptations that are similar to teacher-created adaptations. We have also conducted a qualitative teacher pilot with 12 middle school teachers who teach English language learners. We present a discussion of the tool and pilot.

1. Introduction

The practice of text adaptation is wide-spread among K-12 teachers who teach English language learners (ELLs). This highly labor-intensive practice is necessary, since it is difficult to find authentic texts that are reading-level appropriate for ELLs. Finding appropriate texts becomes especially difficult in classrooms where students have mixed-level reading abilities. In terms of adaptations, research suggests that text summarization, and vocabulary and native language support can facilitate students' reading comprehension and English language skills development [1], [2], [3], [4]. Motivating factors in this research have been: (a) to develop a tool that generates text adaptations that are as effective as teacher-generated adaptations for supporting reading comprehension and English language skills development, and (b) to work with teachers in order to observe their reactions to the tool, and to survey their ideas about appropriate tool use. In light of these goals, we have implemented the Automated Text Adaptation Tool v.1.0 (ATA v.1.0), and have completed a small teacher pilot. Previous research addressing the development of NLP-based reading support tools can be found in [5] and [6].

2. The Automated Text Adaptation Tool (ATA v.1.0)

ATA v.1.0 uses the natural language processing (NLP) capabilities described in this section to automatically generate text adaptations that are similar to teacher-created adaptations. English-based adaptations are helpful for ELLs with any language background, while Spanish features will provide additional help for native Spanish speakers.

2.1. English and Spanish Marginal Notes.

Similar to creating text summaries as adaptations, an automatic summarization tool [7] is used to produce marginal notes in English. The amount of marginal notes generated can be increased or decreased based on students' needs. Using machine translation, Spanish marginal notes can be created for Spanish-speaking ELLs.[1]

2.2 Vocabulary Support

Synonyms for lower frequency (more difficult) words are output using a statistically-generated word similarity matrix [8]. ATA v.1.0 generates *antonyms* for vocabulary in the text using WordNet®: a lexical database containing word sense information for about 150,000 English nouns, verbs, adjectives, and adverbs. For synonym support, [8] was chosen instead of WordNet because it offers broader synonym coverage. *Cognates* are words which have the same spelling and meaning in two languages (e.g., *animal* in English and Spanish). ATA v.1.0 generates these using an ETS English/Spanish cognate dictionary.

2.3 English and Spanish Text-to-Speech (TTS)

The tool offers English and Spanish text-to-speech[2]. English TTS may be useful for pronunciation. Spanish TTS provides access to the Spanish texts for Spanish-speaking ELLs who are not literate in Spanish.

3. Pilot Study with 12 Teachers

Twelve middle school teachers participated in the pilot. The purpose of the study was to collect initial feedback on a) overall tool usability, b) specific feedback on the utility of each tool feature, and c) ideas for tool use. At this early stage, we need to understand how teachers perceive the tool, and to learn which tool uses are aligned with their pedagogical practices.

Teachers were provided with a prototype training manual and access to an online version of ATA v.1.0. They were asked to systematically try out each of the core tool features, and subsequently, to respond to a questionnaire designed to assess their perceptions. Teachers' responses to both the set of 50 Likert-type survey questions, and the five open-ended responses were positive regarding the tool's potential to create accessible texts for their ELLs that would support reading comprehension and English language skills development. Teachers also noted that the tool could be improved by adding an editing capability. This is not unexpected given current limitations of some

[1] See http://www.languageweaver.com.

[2] See http://www.cstr.ed.ac.uk/projects/festival/ & http://cslu.cse.ogi.edu/tts/download/ for details.

NLP capabilities. For instance, we know that statistical synonym identification can generate an incorrect sense of a word for a particular context (e.g., *equipment* is generated for *system*, instead of *organization* in the context of “government.”). The ability to edit system synonym choices would allow teachers not only to remove system errors, and but also to add suitable alternatives. Teachers also offered ideas about best practices for the tool within the school context. They seemed to view the tool either as a *teacher tool for lesson planning*, or as a *student tool to facilitate independent work.*

4. Future Research

We have introduced early work-in-progress for a tool that uses NLP to create text adaptations suitable for ELLs. The teacher pilot outcomes suggest that the tool produces adaptations which have potential as effective support for reading comprehension and English language skills development. Using the tool could save teachers lesson planning time, since much of the adaptation could be automatically generated. Moving forward, we plan to implement suggested tool modifications (e.g., editing function) based on teacher feedback, and we are planning a larger-scale, school-based pilot to evaluate the tool’s utility and effectiveness in terms of measurable learning gains for students' with regard to reading comprehension of content and English language skill development.

Teacher collaboration is essential as we continue to develop the tool. It is important that teachers have a level of comfort and ease with the operation and features of the tool, so that they will be willing to use it themselves and introduce it to their students. Therefore, moving forward, best practices for the tool will be determined in an upcoming larger-scale pilot in collaboration with participating teachers.

References

[1] Carlo, M.S., August, D., McLaughlin, B., Snow, C.E., Dressler, C., Lippman, D., Lively, T. & White, C. (2004). Closing the gap: Addressing the vocabulary needs of English language learners in bilingual and mainstream classrooms. Reading Research Quarterly, 39(2), 188-215.

[2] Echevarria, J., Vogt, M., and Short, D. (2004). Making Content Comprehensible for English Language Learners: the SIOP model. New York: Pearson Education, Inc.

[3] Fitzgerald, J. English-as-a-Second-Language Learners' Cognitive Reading Processes: A Review of Research in the United States, Review of Educational Research, Vol. 65, No. 2 (Summer, 1995), pp. 145-190.

[4] Yano, Y., Long, M. & Ross, S. (1994). The effects of simplified and elaborated texts on foreign language reading comprehension. Language Learning, 44, 189-219.

[5] Schwarm, S. and Ostendorf, M. Reading Level Assessment Using Support Vector Machines and Statistical Language Models. In Proceedings of the Association for Computational Linguistics, Ann Arbor, MI, 523-530.

[6] Heilman, M., Collins-Thompson, K., Callan, J., Eskenazi, M. (2006) Classroom Success of an Intelligent Tutoring System for Lexical Practice and Reading Comprehension. In Proceedings of the Ninth International Conference on Spoken Language Processing. Pittsburgh.

[7] Marcu, D. (2000) The Theory and Practice of Discourse Parsing and Summarization. The MIT Press, Cambridge, Massachusetts.

[8] Lin, D. (1998). Automatic Retrieval and Clustering of Similar Words. In Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics, Montreal, 898-904.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

On Issues of Feature Extraction in Chinese Automatic Essay Scoring System

Tao-Hsing Chang[a,c], Chia-Hoang Lee[c] and Hak-Ping Tam[b]
[a]*Research Center for Psychological and Educational Testing*
[b]*Graduate Institute of Science Education*
National Taiwan Normal University, Taipei, Taiwan, R.O.C.
[c]*Department of Computer Science*
National Chiao Tung University, Hsinchu, Taiwan, R.O.C.

Abstract. Automatic essay scoring is a very important tool for many educational researches. However, the linguistic differences between Chinese and English indicate a need to reconsider various issues when designing Chinese automatic essay scoring systems. This study proposes system architecture for developing a Chinese automatic essay scoring system and analyzes various issues related to feature extraction of Chinese writing.

Keywords. Automated essay scoring, Chinese essay grading, feature extraction.

Introduction

Automatic essay scoring (AES) system is a very important research tool for such areas as educational testing and psychometrics. However, a large number of graded writings is often very difficult to obtain due to the high cost and length of time of human grading. In English, the successful development of AES systems has overcome these limitations, and largely facilitated the progress in these research areas. By contrast, the lack of Chinese automatic essay scoring systems (CAES) has limited the scale, quantities, and validity of these research areas.

The linguistic difference between Chinese and English indicates the need to reconsider various issues when designing CAES. This study proposes a system architecture for CAES, and analyzes various issues related to feature extraction in Chinese writing. The designing consideration is to incorporate the distinct writing factors in Chinese language into the basic architecture of the available English AES.

1. Architecture for Chinese Automatic Essay Scoring System

The proposed architecture for CAES consists of three phases, namely preprocessing, feature extraction and machine learning. The preprocessing phase focuses on the lowest level of essay analysis, namely word segmentation and unknown word extraction, and part-of-speech tagging. The feature extraction phase includes the design of features categories corresponding to writing requirements and skills, as well as algorithmic methods to identify and extract the features. The machine learning phase refers to constructing or learning a statistical model or intelligence model consistent with the training data, and thus to scoring essays based on the model.

The preprocessing phase is required for CAES, although it may be not in English AES. Many previous works [2], including ours [7], have addressed the modules in the phase. The machine learning phase contains discretization, feedback and prediction

modules. The discretization module discretizes feature values, and maps feature values into different grading levels. These values serve as the input values for both the feedback and prediction modules. The feedback module analyzes the shortcomings of an essay, and gives a critique. A prediction module, such as linear regression and Bayesian method, is used to score an essay based on its features. The task of designing a good prediction module is also an important question.

The feature categories proposed for the feature extraction phase consist of five modules, namely sentence structure, topics, organization, figure-of-speech and surface. The surface features including the number of words, the average length of sentences are also included due to a high correlation with scores. The next section discusses these features in detail.

2. Issues of Feature Extraction Module

According to the above proposed CAES architecture, feature selection is a very important task, because it significantly affects the system performance. Many previous studies [1][4][5] in contrast rhetoric and contrast linguistics focus on the difference in writing features between Chinese and English. These studies focus on four directions covering sentence structure, topics, organization, and figures of speech. These issues are discussed separately below.

Many studies have observed that sentence strcuture is more flexible and loose in Chinese than in English. Jaio [9] noted three fundamental differences. First, the sentence structure in English can be perceived as a tree structure, while that in Chinese can be perceived as a linear structure. Second, sentences in English emphasize hypotaxis, which refers to grammatical subordination, while sentences in Chinese focus on parataxis, which refers to grammatical coordination. Third, a sentence in English is complete as long as the subject and predicate appear. In contrast, the amount of information in Chinese sentence is not strictly constrained. Lee and Zeng [11] pointed out that the ordering of three basic elements (subject, verb and object) and the type of tags show no fundamental difference in either Chinese or English discourse. However, the constituents may be very different. For example, verbs and adjectives can be used as sentence subjects in Chinese, but have to be changed respectively into infinitives or participles before they can be used this way in English. The same restriction also appears in the predicates. The above discussion demonstrates that developing a powerful parser for Chinese is a quite difficult task.

One writing requirement is to select materials, where the illustration is consistent with the topic. However, the selection of material is quite different in Chinese and English, due to differences between Eastern and Western cultures. Chinese students often quote authoritative argument and classical articles, and seldom express personal opinions and feelings. The use of rhetoric is quite indirect and moderate. In contrast, western students like to present evidence to support their arguments or viewpoints. Critical and logical discussions are quite important in English writing. This observation indicates that the material selections of a Chinese essay should not be judged by the English automatic scoring system, because of these different perspectives.

Many studies have noted fundamental differences in discourse organization between Chinese and English writing. In particular, Kaplan [10] pointed out that the organization of discourse in English writing often appears as a linear sequence. In contrast, the discourse organization in Chinese writing often has a spiral pattern. In [11], the tree structure of discourse also shows that the coherence of discourse in English tends towards the relationship between main and subordinate clauses. In contrast, the

structure of Chinese discourse tends towards parallel sequences. Scollon *et al.* [3] also observed that people in western cultures use a deductive method of reasoning or argument, while people in eastern cultures use inductive reasoning. This finding indicates that human graders from different cultures may give different judgments for topic organization.

The usage of figures of speech (FOS) is an important factor for high quality of writing. Although Chinese and English writing both have many common usages of FOS, each language has its own unique FOS. Bai and Shi [6] found that 16 FOS used in Chinese and 26 FOS employed in English each included 6 FOS not observed in any other language. Even similar FOS is used differently in each language. Hence, although graders use FOS to judge the essay, the factors may be different.

3. Prototype of CAES

A prototype [8] of CAES was implemented based on the proposed architecture and above discussion. In the preprocessing phase, the prototype utilizes a method proposed in our earlier research [7] for extracting word segmentation and POS tagging. The feature extraction phase only uses the surface and figure of speech modules to show the performance of the proposed prototype. The FOS module contains three FOS: "pai-bi", "pi-yu" and literary "pi-yu". The machine-learning phase of the prototype employs the ID3 decision tree [13] as a prediction module. Experimental results in [8] indicate that any usage of a single FOS can improve the performance of CAES. Furthermore, the accuracy and exact rates of scoring essays is 0.91 and 0.48 respectively when three FOS features were employed. The results show that the prototype performs well despite implementing only a few features of Chinese.

This study discusses various design factors, and proposes suggestions for designing CAES. The proposed system architecture based on the discussions will be useful as a reference for efficiently designing and developing a CAES system.

Acknowledgment. This work was supported in part by the National Science Council, Taiwan, under Grant NSC 94-2213-E-009-140 and CREED, National Taiwan Normal University, under Grant 95E0012-A-05

References

[1] *A contrastive study of English and Chinese writing*, J. G. Cai, Fudan University Press, 2001.
[2] *Chinese part-of-speech analysis*, CKIP, Technical Report 93-05, Academia Sinica, Taipei, 1993.
[3] *Contrastive discourse in Chinese and English: A critical appraisal*, R. Scollon, S. W. Scollon & A. Kirkpatrick, Foreign Language Teaching and Research Press, 2000.
[4] *Contrastive studies of figures of speech in English and Chinese*, G. N. Lee, Fujian People's Publishing House, 1999.
[5] *Contrastive study of writing & rhetoric in English and Chinese*, J. G. Cai, Fudan University Press, 2003.
[6] S. T. Bai & A. W. Shi, A Comparative Study of Figures of Speech between Chinese and English, *Journal of Xinzhou Teachers University* **18:1** (2002),70-71.
[7] T. H. Chang & C. H. Lee, Automatic Chinese Unknown Word Extraction Using Small-Corpus-based Method, *IEEE NLP-KE '03*, 459-464, Beijing, China, 2003.
[8] T. H. Chang & C. H. Lee, Enhancing Automatic Chinese Essay Scoring System from Figures-of-Speech, *PACLIC 20*, 28-35, Wuhan, China, 2006.
[9] C. Y. Jiao, A syntactic comparison and transformation between English and Chinese, *Journal of Yancheng Teachers College* **22:2** (2002), 83-87.
[10] R. B. Kaplan, Cultural Thought Patterns in Intercultural Education, *Language Learning* **16** (1966), 1-20.
[11] X. L. Lee & K. Zeng, Heterogeneity and Homogeneity of Sentence Structure in English and Chinese, *Journal of Shenyang University* **12:1** (2001), 52-55.
[12] L. J. Liu, A contrastive study of Discourse Structure in English and Chinese, *Modern Foreign Language* **86:4** (1999), 408-419.
[13] J. R. Quinlan, Induction of Decision Tree, *Machine Learning* **1:1** (1986), 81-106.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Domain-Specific and Domain-Independent Interactive Behaviors in Andes

Min CHI and Kurt VANLEHN
Learning Research and Development Center,
University of Pittsburgh, Pittsburgh, PA 15260 U.S.A

Abstract. In this study, students studied two different domains in the same Intelligent Tutoring System, Andes. Analysis of 435 log files from 22 subjects indicated that there are two types of interactive behaviours in Andes: domain-independent and domain-specific. We believe the existence of the domain-specific behaviours is one possible reason that similar meta-cognitive behaviors has not been found across domains in a single ITS curriculum [1][2]. This paper describes the study and the potential applications of these findings.

Introduction

In this study, we ported our existing physics tutor Andes [4] to probability by replacing the physics knowledge base with the probability concepts and principles [3]. Apart from the declarative domain knowledge, the two Intelligent Tutoring Systems (ITSs) were identical. We refer to them as Andes-physics and Andes-probability respectively. An important question is what happens when two different task domains exist in the same interface? Would students engage in similar interactive behaviors or different ones? In this paper, we will use the term interactive behaviors refer to the individual actions that the students performed on the ITS.

Previous studies in this area have shown that students do *not* exhibit the same patterns of meta-cognitive behaviors across domains in a single ITS curriculum [1][2]. Baker et al. built a gaming detector by aggregating patterns of students' behavior from logs. They investigated how well it transferred across domains in the same ITS within a single student population. They found that the detector built from data in one domain does well within the domain but cannot successfully predict behavior across domains.

In this study, we used aggregate measurements of interactive behaviors, such as the total time taken or the total number of entries made. Two types of behaviors were defined: domain-specific and domain-independent. An interactive behavior is called domain-specific if its appearance was associated with the domain characteristics and thus its appearance in one domain did *not* predict its appearance in another domain; whereas it is called domain-independent if its appearance was *not* associated with the domain characteristics and its appearance in one domain did predict its appearance in another. We found that both types of behaviors existed in Andes. Next, we will describe this study in a greater detail. Following that we will present the results. Finally, we presented the conclusions.

1. Methods

Data was collected from 23 students during the fall of 2005 and the early spring of 2006. Students were required to have basic knowledge of high-school algebra but to not have taken college-level statistics or physics courses. They studied two domains: probability and physics. Each domain contained ten principles. Students first took a background survey and then went through probability instruction followed by physics instruction. Each instruction consisted of four steps: 1) pre-training, 2) pre-test, 3) training on Andes, and 4) post-test. During the pre-training, the students read a general description of each principle, reviewed some examples, and solved some problems.

During training on Andes, students first watched a video of a problem being solved in Andes. Then they solved twelve probability problems in Andes during the probability instruction and eight physics problems during the physics instruction. Andes gives immediate feedback as soon as an action is performed. Entries are colored green if they are correct and red if incorrect. We distinguish between two types of errors: *tiny errors* and *red errors*. *Tiny errors* are those likely due to lack of attention such as typos. *Red errors* are likely caused by conceptual misunderstandings or a lack of domain knowledge such as incorrect principle applications. When *tiny errors* occurred, Andes popped up an automatic error message. When *red errors* occurred, Andes colored the entry red and gave the students the option to ask for *what's-wrong help* or fix it on their own. Students could ask for *next-step help* at any time if they are not sure what to do next. Both *what's-wrong help* and *next-step help* were provided via a sequence of hints that gradually increased in specificity. The last hint in the sequence, called the *bottom-out hint*, told the student exactly what to do.

2. Results

One student was eliminated due to a perfect probability pre-test score. For the remaining 22 students, there were 435 Andes log files, 259 for probability and 176 for physics. For each log file, we extracted seven interactive behaviors: the total time and the total number of next-step help, what's-wrong help, bottom-out hint, red errors, tiny errors, correct entries, and total entries. Table 1 shows that students had significantly higher values during the physics training than during the probability training on all behaviors. This suggested that the physics training problems seemed to be more difficult and challenging for students than the probability ones.

Table 1. Compare various interactive behaviours across domains

	Next-Step	Bottom-Out	What's-Wrong	Red Errors	Tiny Errors	Correct Entries	Total Entries	Total time
Probability	16.00	3.02	1.05	4.40	2.05	14.94	17.98	793.4
Physics	34.67	7.85	5.53	11.08	4.41	29.1	39.33	1351.

Table 2 shows the correlation of the interactive behaviors across the instruction sessions in Andes. The unit of analysis was the student. Each correlation is the result of 22 2-tuples, one pair for each student. We found that two help-seeking behaviors, *next-step help* and *bottom-out hints*, were correlated across domains while *what's-wrong help* was not. These results make sense: asking for *next-step help* may result from the students' lack of general problem-solving skills rather than any domain-specific characteristics. Since the last hint in the sequence of *next-step help* is a *bottom-out hint*, it was no surprise that *bottom-out hints* were also correlated across domains. Asking for *what's-wrong help*, on the other hand, is more likely to be driven by the domain

characteristics. We predict that students' ability or willingness to fix an error on their own is largely dependent on how difficult they think the domain is. Since physics appeared be much harder for them than probability, students were more likely to ask for *what's-wrong help* in physics than in probability.

We interpreted the correlation between the number of *red errors*, correct entries, total entries, and the total time as a sign that students' problem-solving strategies and general competence remained consisted across domains. It indicated that Andes did not teach them anything about general problem solving skills. The lack of correlation of making tiny errors is likely due to domain complexity: for example, students often need to input both a number and a unit in physics but only a number in probability.

Table 2. Correlation of interactive behaviours across Probability and Physics

#	Interactive Behaviours	correlation	#	Interactive Behaviours	correlation
1	Next-Step Help	$r= 0.8015, p< 0.0001$	**5**	Tiny Errors	-
2	Bottom- Out Hints	$r= 0.6684, p=0.002$	**6**	Correct Entries	$r= 0.719, p=0.001$
3	What's- Wrong help	-	**7**	Total entries	$r= 0.5629, p=0.012$
4	Red Errors	$r=0.545, p=0.016$	**8**	Total time	$r=0.834, p< 0.00001$

3. Conclusions and Discussions

Apparently, our findings seem to contradict previous studies [1][2] in that some meta-cognitive behaviors in our study seem stable across domains. However, we argue that our results are consistent with theirs. Baker et al. aggregated multiple meta-cognitive behaviors with distinct patterns of activities whereas we focused on isolated but still interpretable activities. In order for their predictor to function accurately, the relative frequencies of each meta-cognitive behavior would need to remain constant across domains. That is, all of the individual behaviors in their study would need to be domain-independent. However, we have shown that some behaviors that matter to their detector are domain-specific. For example, one crucial gaming criteria in their study was that students who keep making errors do eventually get the correct answer. This would show up in our study as students keep asking for *what's-wrong help* and eventually get the correct answer. We expect that some of behaviors in their study would be domain-specific. In this study, we investigated the interactive behaviors in only two domains: probability and physics. In order to draw a more comprehensive conclusion, we would need to test it in multiple domains. We think it would be worth the attention and time because we expect that a model constructed solely on domain-independent behaviors would exhibit greater prediction accuracy across domains and thus lead to a more accurate student model. We believe that porting an existing ITS to a new domain is not only efficient in terms of time and cost [3], but also may improve learning by allowing more accurate prediction and prevention of help-abusers.

References

[1] Baker, R.S.J.d., Corbett, A.T., Koedinger, K.R., Roll, I. *Generalizing Detection of Gaming the System Across a Tutoring Curriculum*. Proceedings of the 8th ITS (2006). 402-412

[2] Baker, R.S., Corbett, A., Koedinger, K., Roll, I. (2005) *Detecting When Students Game The System, Across Tutor Subjects and Classroom Cohorts* . Proceedings of User Modeling 2005, 220-224.

[3] Chi, M, and VanLehn K, Porting an Intelligent Tutoring System across Domains. 13th International Conference on Artificial Intelligence in Education

[4] VanLehn, K., Lynch, C., Schultz, K., et al. (2005). The Andes physics tutoring system: Lessons learned. *International Journal of Artificial Intelligence and Education, 15*(3), 147-204.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Porting an Intelligent Tutoring System across Domains

Min CHI and Kurt VANLEHN
Learning Research and Development Center,
University of Pittsburgh, PA 15260 U.S.A

Abstract. One possible approach to reducing the cost of developing an intelligent tutoring system (ITS) is to reuse the components of an existing ITS. We used this approach to develop an Andes probability tutoring system by modifying the declarative knowledge of the Andes physics tutoring system. We claim that if we cluster various educational domains into groups based on their problem-solving methods [2], then it will be more efficient to port an existing ITS to a new domain in the same cluster than to build a new ITS from scratch.

Introduction

One of the main obstacles to deploying a new ITS is the significant time and cost requirements. A diverse team of computer programmers, domain experts, and educational theorists, are needed during the entire process of development and testing. Two approaches have been taken to reduce such costs. One is to use authoring tools which provide a relatively simple development environment to reduce the programming load [1][4]. Another is to reuse shared components from preexisting systems. However, little effort has been spent on the latter despite the fact that a large number of components are shared across ITSs. One of the reasons for this is that educational research is often domain-specific. Research on effective physics instruction is often conducted independently from research on probability instruction. There is often little consensus on the effectiveness of an instructional method within a single domain let alone across domains. As a result, there are no universal standards for tutoring components and most ITSs are entirely domain specific.

However, we believe that it is more efficient in terms of time and cost to port an existing ITS to a new domain than to build a new system from scratch if the two domains share the same problem-solving method (PSM) [2]. We ported our existing Andes physics tutor to a new domain, probability, by replacing the declarative knowledge within the existing physics knowledge base with new probability principles and concepts. All of the procedural knowledge, help procedures, and interface components were retained. We will describe this process in greater detail below.

1. Porting Andes from Physics to Probability

Andes is an ITS that has helped hundreds of students improve their understanding of college physics [5]. It consists of five components: a Knowledge Base (KB), a Pedagogical Module, a Problem Solver, a mathematics package, and a GUI. The KB comprises declarative domain knowledge such as principles, concepts, and problem definitions. The Pedagogical Module comprises procedural knowledge including optimal problem solving methods and instructional algorithms for the provision of hints, which include procedures for determining how students should solve problems

and algorithms determining the types of hints provided to students and their level of specificity. The Problem Solver employs inference rules to automatically derive a solution graph for each problem. Each graph represents multiple solutions to the given problem. The Mathematics Package handles algebraic simplification. The GUI permits students to solve problems in a manner similar to using a pencil and paper. These components are described in much greater details in [5].

We ported Andes to a probability domain by replacing the physics KB with declarative knowledge for probability. The new KB consists of seven concepts, 10 principles (see Table 1), and 36 problem definitions. The seven concepts were: *sample space* (Ω), *single event* (X), *event complement* (~X), *intersection* ($X \cap Y$), *union* ($X \cup Y$), *conditional event* (X|Y), and *probability* (p). For each principle, we wrote three levels of hints: *pointing*, *teaching* and *Bottom-Out*. For example: for the "Addition Theorem for two events, X and Y", the pointing hint is "*Please apply the addition theorem for two events X and Y*"; the teaching hint is "*The definition of addition theorem for two events A and B is $P(A \cup B)=P(A)+ P(B)- P(A \cap B)$. Please apply the addition theorem on event X and Y.*"; and the bottom out hint is "*Write the equation $P(X \cup Y) =P(X) + P(Y) - P(X \cap Y)$*".

Table 1. Major Principle of Probability

Complement Theorem	Mutually Exclusive Theorem
Addition Theorem for two events	Addition Theorem for three events
De Morgan's Theorem	Independent Events
Conditional Probability	Conditional Independent Events
Total Probability Theorem	Bayes Rule

For each of the 36 problems, the Andes' Problem Solver automatically produced a solution graph. The number of principle application required to solve a problem varied from 1 to 9. Before Andes-probability was ready for testing, we also asked one of Andes' developers to set aside one afternoon to develop the one GUI component necessary for defining the probability of an event on Andes. Altogether, it took a moderately experienced ITS developer, the first author, roughly 10 days and an experienced Andes programmer less than one day to port Andes from physics to probability . Figure 1 shows the Andes-probability GUI.

There are three primary reasons why the port was so simple. First, although probability and physics are dissimilar domains at the surface level, they share PSMs which are normally described informally as ways to solve a task [2][3]. In our case, the two domains are both principle-based domains and problem solving in them involves producing a set of equations that solve for a sought quantity. Since Andes' Problem Solver is independent from its KB, it can also be used to automatically produce solution graphs for the probability problems. Generally speaking, producing these solution graphs is one of most expensive tasks when deploying an ITS. Second, the remaining components which include the Pedagogical Module, the mathematics package and the GUI, are also domain-independent and thus can also be reused. For example, the Pedagogical Module includes the domain-independent pedagogy rules such as "variables must be defined before being used". At the same time, various Andes hints can be extracted from the declarative KB instead of being embedded in the Pedagogical module. Finally, we also reused Andes' error handlers. Andes distinguishes between two types of errors for error handling: *tiny errors* and *red errors*. *Tiny errors* are those

that are likely due to lack of attention such as typos. *Red errors* are those that are due to conceptual misunderstandings or a lack of knowledge. *Tiny errors* are largely domain-independent and thus we did not need to rewrite anything for them. *Red error* handling is primarily driven by the declarative KB which we had already rewritten so no additional development was required.

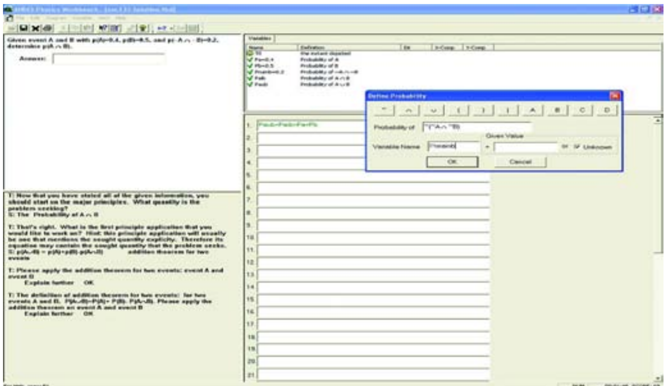


Figure 1. Andes' Interface for Probability

2. Conclusions

Developing high-level authoring system have been proven to be general and successful, e.g. the Cognitive Tutor Authoring Tools [1] and REDEEM [4], but the cost of producing them is itself very high. In this paper, we have described how porting an existing tutor to new domains might be more efficient in terms of time and cost. However, it is a less general alternative. We believe there are two essential requirements for porting to be efficient: first, the declarative KB should be constructed independently from the rest of components in the ITS; second, the two domains should share a common PSM. If either requirement is violated, we expect the cost of porting to increase radically, potentially exceeding the cost of starting from scratch.

Clearly, there are additional factors, such as cognitive similarity that will also affect the effectiveness of the porting. Those factors should be explored in future work. One way to make our approach more doable for other domains or to make this kind of technology migration easier is for the AI&Ed field to cluster various domains into groups based on common PSMs, build a digital library of ITSs, and set up standards for documentation. This research can be guided by preexisting work on the development of reusable components for knowledge-based systems [3].

References

[1] Aleven V, Bruce M. McLaren, Jonathan Sewall, Kenneth R. Koedinger (2006) The Cognitive Tutor Authoring Tools (CTAT): Preliminary Evaluation of Efficiency Gains In the Proceedings of the 8th International Conference on Intelligent Tutoring Systems Jhongli, Taiwan, June 26-30, 2006

[2] Chandrasekaran, B "Towards a Taxonomy of Problem Solving Types" AI Magazine 4(1) 1983, 9-17.

[3] D. Fensel and E. Motta. Structured Development of Problem Solving Methods. IEEE Transactions on Knowledge and Data Engineering, 13(6):913–932, 2001.

[4] Major, N. 1995. REDEEM: Creating Reusable Intelligent Courseware. In Proceedings of the International Conference on AIED., Greer, J., ed., AACE, Charlottesville, VA, pp. 75-82.

[5] VanLehn, K., Lynch, C., Schultz, K., Shapiro, J. A., et al. (2005). The Andes physics tutoring system: Lessons learned. International Journal of Artificial Intelligence and Education, 15(3), 147-204.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Learning from Collaboratively Observing Videos during Problem Solving with Andes

Scotty D. CRAIG, Kurt VANLEHN, Soniya GADGIL, & Micki CHI
University of Pittsburgh, LRDC
3939 O'Hara Street, Pittsburgh, PA 15260, USA
scraig@pitt.edu

Abstract. Learning by observation has long been a traditional method of learning. Recent work has pointed toward collaboratively observing tutoring as a promising new method for observational learning. Our current study tested this new method in the PSLC physics LearnLab where students were introduced two topics of rotational kinematics by observing videos while problem solving in Andes. The students were randomly assigned to a pair condition that collaboratively observed a video of an expert tutoring or providing an example, or to a solo condition that observed a video of an expert worked example. Several robust and normal learning measures were collected, however, to date only multiple choice measures have been analyzed. Students' performance on the multiple choice questionnaires revealed significant pretest to posttest gains for all conditions. However, no differences have been found among conditions for normal learning measures.

Learning from observation is a common and natural method of human learning. Observation has long been seen as a key learning method for children when acquiring basic tasks such as language and cultural norms [7]. Within psychology, research on human learning from observation dates to the preliminary work by Bandura [1] and over the years has been referred to under various titles such as observational learning, social learning, and vicarious learning (e.g., [5]). Observational learning has been investigated in various settings such as using real time observations, video tapes of humans and cartoons, and listening to audiotapes ([1]).

Observing tutoring has recently emerged as promising new focus in the literature on observation ([2] [3]). By observing tutoring, we mean that a learner views and overhears the dialogue between a tutor and a tutee. If students can learn as effectively from observing tutoring as from interacting with a tutor, then we would have an effective alternative for human tutoring, which is labor-intensive, and for intelligent computer tutoring (ITS), which is expensive to develop.

The evidence for the benefit of observing tutoring has been mixed, but systematic. Craig et al. ([3] Experiment 1) contrasted pretest to posttest gains of low-knowledge learners (tutees) who interacted directly with AutoTutor [6] to that of yoked low-knowledge solo observers. AutoTutor is a natural-language ITS that tutors 12 computer literacy topics. As tutees interacted with AutoTutor, the visual and auditory contributions of AutoTutor and the contributions of each learner were recorded. Students in the observing condition were shown these recordings. The results of immediate pretest to posttest showed that tutees significantly outperformed observers.

In a second experiment, Craig et al. ([3], experiment 2) attempted to improve learning by adding a third collaboratively observing condition because having the

capability to collaborate has been shown to improve learning and provide a deeper understanding of the material (See [4]). In the collaborative condition, two participants together at a computer monitor watched the video of an interactive AutoTutor session on computer literacy. The collaborative observers were encouraged to pause the video and talk to each other about information in the video. As in the first experiment, immediate pretest to posttest results showed once again the tutees outperformed the solo observers. While the collaborative observers' gains showed improvement over the solo observers, the difference was not significant enough to become reliable. However, the collaborative observer gains were elevated to the point that they were no longer significantly lower than the tutees' gains.

Chi et al. [2] investigated learning from observing tutoring using an expert tutor instead of an ITS and a more complicated topic, solving physics problems. The study compared the learning outcomes of five different instructional conditions; the three that are pertinent to this paper are: one-to-one tutoring with an expert tutor, observing tutoring collaboratively, and observing tutoring alone. Ten expert tutoring sessions were recorded, then the videos were studied either by pairs of students or by students working alone. Unlike the Craig et al. experiments, all participants in this study also solved the problems as they were observing them being solved. The collaborative observers' gains were not significantly different from the tutees' gains, and both were significantly greater than the solo observers' gains.

Although this is exactly the same pattern of results as in the Craig et al. ([3], Exp. 2), Chi et al. found that collaboration did improve learning for collaborative observers over solo observers. One explanation for this discrepancy is the amount of collaboration. A follow up analysis of audio tapes of the Craig et al. Experiment 2 revealed that collaborative observers found that they had an average of 2.91 conversational turns per session with an average session lasting 35 minutes, whereas the collaborative observers in the Chi et al. study produced on average 121.22 conversational turns per 35 minute interval. The increased level of collaboration was most likely due to the task demands of the study. While the Craig et al. study did not require learners to perform any task other than watching the video, learners in the Chi et al study performed a problem solving task while observing tutoring.

Chi et al. divided tutees into high ability and low ability, based on a median split on their pretest scores. The video tapes of the high ability tutees produced more gains from the collaborative observers than the video tapes of the low ability tutees. Because the videos of the high ability tutees had fewer tutee errors and remediation episodes than videos of the low tutees, this may have raised their effectiveness as scaffolding instruments. This suggests that a video that has no errors at all might be even more effective still. Such a tape would be essentially a worked example—a demonstration of a correct solution to the problem.

The study reported here took this methodology into the classroom and tested the scaffolding explanation. In doing so, we compared collaborative observers of tutoring video (Collaboratively observing tutoring condition) to collaborative observers of a worked example video (Collaboratively observing examples condition). A third condition, Solo observing examples condition, was comprised of solo observers of a worked example video. Since Chi et al. [2] did not find learning gains for individuals observing tutoring, the solo observing of tutoring condition was not taken into the classroom in order to avoid exposing students to an ineffectual learning condition.

United States Naval Academy (USNA) students (Age: 18-19; N = 68) from three sections of the PSLC physics LearnLab (www.learnlab.org) were randomly assigned to

one of three conditions described above. They solved rotational kinematics problems using the Andes tutoring system while simultaneously watching a video showing the same problems being solved. A problem solving task during observing was chosen in order to increase active engagement with the task as seen in the Chi et al study. The video showed either a tutorial dialogue with an expert physics tutor assisting a student who was solving Andes problems, or a worked example with the same expert solving the same Andes problems while explaining the steps orally. This acquisition phase occurred before the material was covered in the course. The students were assessed individually immediately after training using two multiple-choice tests that required knowledge transfer and three Andes transfer problems on rotational kinematics. The two multiple-choice tests were alternated between students as pretest and posttests. Long term measures of Andes homework problems were also collected.

At this writing, only the multiple-choice tests have been analyzed. They show that all students gained significantly from pretest to posttest, $F(1, 65) = 14.987$, $p < .001$. However, there were no significant differences among conditions on the multiple choice tests. That is, the collaborative observers of tutoring, the collaborative observers of worked examples, and the solo observers of worked examples all seem to have learned the same amount. This lack of significance did not appear to be due to a ceiling effect among the conditions given the means (See Table 1).

Table 1. *Means and standard deviations for pretest, posttest and gain scores for the three conditions*

Condition	Pretest		Posttest		Gain scores	
	M	SD	M	SD	M	SD
Solo observing examples	.58	.23	.69	.18	.11	.20
Collaboratively observing examples	.57	.22	.65	.20	.08	.19
Collaboratively observing tutoring	.58	.19	.67	.17	.08	.16

This evidence would seem to support the Craig at al finding that there is no benefit for observing tutoring collaboratively. However, this interpretation is only based on the multiple choice data and thus is still speculative at this point. This pattern could change when the learners process data from the task (e.g. level of collaboration, task engagement or video use) and long-term measures from the homework are analyzed.

Acknowledgments. This project was supported by NSF award SBE-0354420 to the Pittsburgh Science of Learning Center. The authors of this paper would like to thank the PSLC Physics LearnLab for their help and support on this project.

References

[1] Bandura, A. (1986). *Social foundations of thought and action: A social cognitive theory.* Englewood Cliffs, NJ: Prentice Hall.

[2] Chi, M. T. H., Hausmann, R., & Roy, M. (under review). Learning from observing tutoring collaboratively: Insights about tutoring effectiveness from vicarious learning. *Cognitive Science*.

[3] Craig, S. D., Driscoll, D., & Gholson, B. (2004). Constructing knowledge from dialogue in an intelligent tutoring system: Interactive learning, vicarious learning, and pedagogical agents. *Journal of Educational Multimedia and Hypermedia, 13*, 163-183

[4] Dillenbourg, P. (1999). What do you mean "collaborative learning"? In P. Dillenbourg (Ed.), *Collaborative learning: Cognitive and computational approaches* (pp. 1-19). Oxford: Elsevier.

[5] Gholson, B., & Craig, S. D. (2006). Promoting constructive activities that support learning during computer-based instruction. *Educational Psychology Review, 18*, 119-139.

[6] Graesser, A.C., McNamara, D.S., & VanLehn, K. (2005). Scaffolding deep comprehension strategies through Point&Query, AutoTutor, and iSTART. *Educational Psychologist, 40*, 225-234.

[7] Rogoff, B., Paradise, R., Mejia Arauz, R., Correa-Chavez, M., & Angelillo, C. (2003). Firsthand learning through intent participation. *Annual review of psychology*, 54, 175-203

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Linking Educational Materials to Encyclopedic Knowledge

Andras CSOMAI and Rada MIHALCEA [1]
University of North Texas

Abstract. This paper describes a system that automatically links study materials to encyclopedic knowledge, and shows how the availability of such knowledge within easy reach of the learner can improve both the quality of the knowledge acquired and the time needed to obtain such knowledge.

Introduction

According to studies in cognitive science [1,6], an important aspect of the understanding and learning process is the ability to connect the learning material to the prior knowledge of the learner. Similarly, research in active reading and learning [3,4,5] has shown that active reading skills can improve the effectiveness and efficiency of information comprehension.

The amount of background knowledge necessary for a satisfactory understanding of an educational material depends on the *level of explicitness* of the text. However, it is almost impossible to create pedagogical materials that simultaneously serve the needs of both low- and high-knowledge users. Additionally, although the proactive construction of knowledge through active reading was found to increase the effectiveness of the learning process, such skills are not very advanced in many students [3]. It is therefore desirable to develop methods that can seamlessly integrate additional information in the learning material – such that low-knowledge readers can have immediate access to additional explanatory information that can facilitate a deeper understanding of the study material, and they can consequently develop active reading skills.

In this paper we present how our *Wikify!* system [2] can be used to improve educational materials by automatically selecting keywords, technical terms and other key concepts and linking them to an external knowledge source, in our case an online encyclopedia (Wikipedia). This can be thought of as an artificial extension of the reader's knowledge-base that can be accessed on a per-need basis, and is expected to be particularly useful for the increasingly popular online learning environments, where the lack of a direct communication between the instructor and the student can deprive the learner from the ability to quickly receive answers to questions related to background knowledge.

1. Wikipedia and Text Wikification

Wikipedia (`http://en.wikipedia.org`) is an online encyclopedia that has grown to become one of the largest online repositories of encyclopedic knowledge, with millions of

[1] Correspondence to: Rada Mihalcea, University of North Texas, P.O. Box 311366, Denton, TX. E-mail: rada@cs.unt.edu

articles available for a large number of languages. In fact, Wikipedia editions are available for more than 200 languages, with a number of entries varying from a few pages to more than one million articles per language.

Given a text or hypertext document, we define "text wikification" as the task of automatically extracting the most important words and phrases in the document (keywords), and identifying for each such keyword the appropriate link to a Wikipedia article with detailed explanatory information about the corresponding keyphrase.

Text wikification is performed in two steps. The first step consists of identifying those words and phrases that are considered important for the text at hand. These typically include technical terms, named entities, new terminology, as well as other concepts closely related to the content of the text. This task is similar to the well studied problem of *keyword extraction*, and correspondingly the *Wikify!* system uses state-of-the-art keyword extraction techniques coupled with novel features specific to online encyclopedias.

The second step consists of finding the correct Wikipedia article that should be linked to a candidate keyword. Here, we face the problem of link ambiguity, meaning that a phrase can be usually linked to more than one Wikipedia page, and the correct interpretation of the phrase (and correspondingly the correct link) depends on the context where it occurs. This task is in fact analogous to the problem of *word sense disambiguation* and our system again uses state-of-the-art techniques to address this problem.

The *Wikify!* system creates wikified documents hardly distinguishable from those created by human annotators [2], allowing us to automatically generate high quality hyperlinked texts.

2. Wikification: A History Test

The hypothesis guiding our experiment is that the *Wikify!* system can facilitate the access to information by students, and consequently it can improve the learning process. In order to evaluate this hypothesis, we devised a test that simulates the situation where a student requires additional related information to understand a certain study material.

We created a test by selecting 14 questions from a quiz from an online history course taught at the University of North Texas. The questions were selected so that they inquired about encyclopedic information rather than complex analyses or inferences about a problem. All the questions consisted of multiple choice test items, with a relatively high-degree of difficulty – meaning that in most cases common knowledge was not enough to provide an answer. Out of the 14 questions, half were wikified (including the answer choices), and half were left in their original format (raw text).

60 students were asked to participate in the experiment. Each participant was presented with the set of 14 questions, where randomly either the first seven or the last seven were automatically wikified with the *Wikify!* system. Overall, any single question was presented to an equal number of students in a wikified and non-wikified version. Given the size of the experiment, this setting allows us to compensate for the individual abilities of the participants as well as for the various degree of difficulty of the questions.

All the participants received the same set of instructions, which asked them to answer all the questions in one sitting. The instructions specifically indicated that the students were allowed to use *any* source of information they had access to,[2] including their own knowledge, search engines, or the Wikipedia links provided inside the questions. Students were not required to use the Wikipedia links available in the wikified questions, nor were they

[2]Note that the students taking the original quiz during the history course itself were also allowed to use any additional information they needed in order to answer the quiz questions.

prohibited to use the Wikipedia itself for answering the non-wikified questions. When an answer could not be found, the students were also given the possibility to skip the question.

After the test answers were submitted by all the participants, we post-filtered the submissions by removing all the tests left unfinished (13 tests fell under this category), as well as the tests where all the questions were skipped (3 tests). Both these conditions were interpreted as lack of seriousness, and therefore the corresponding tests were discarded. This left us with a final set of 44 complete tests, leading to a total of 44 answers for each of the 14 questions, accounting for 22 answers collected for the wikified version of the questions, and 22 for the non-wikified version.

We were primarily interested in finding the answer to two main questions: (1) Does the immediate access to relevant encyclopedic information improve the quality of the knowledge acquired for a specific topic? and (2) Does bringing such additional information within easy reach reduce the time required to acquire knowledge?

To answer the first question, we determined the number of questions correctly answered using the wikified version versus the non-wikified one. To answer the second question, we determined the time spent to answer the questions for each of the two versions. Table 1 shows the average calculated over the 22 answers collected for each version for each of the 14 questions.[3]

	Wikified	Non-wikified
CORRECTNESS (%)	78.89	75.97
TIME (SEC.)	62.70	70.33

Table 1. Evaluation results for wikified versus non-wikified test items.

In terms of correctly answered questions, the wikified version of the test items resulted in a relative error rate reduction of 12.2% with respect to the non-wikifed version ($p < 0.1$, paired t-test). Moreover, the time required to collect the required knowledge was also significantly reduced in the case of the wikified questions, with a relative time reduction of 25.5% ($p < 0.05$, paired t-test).

Both questions were answered positively: the immediate access to relevant encyclopedic information seems to improve both the quality of the knowledge acquired, and the time needed to obtain such knowledge. While we do not wish to discuss here the reliability of the information contained in Wikipedia, we believe that these results suggest that providing students with information relevant to the topic of study, and bringing such information within easy reach through hyperlinking, are both successful strategies for increased effectiveness in pedagogical tasks.

References

[1] Walter Kintsch. *Comprehension: A paradigm for cognition*. Cambridge Uniersity Press, New York, United States of America, 1998.

[2] Rada Mihalcea and Andras Csomai. Wikify! linking documents to encyclopedic knowledge. *(Under Review)*, 2007.

[3] T. Murray. Metalinks: Authoring and affordances for conceptual and narrative flow in adaptive hyperbooks. *International Journal of Artificial Intelligence in Education*, 13(1), 2002.

[4] C. Roast, I. Ritchie, and S. Thomas. Re-creating the reader: Supporting active reading in literary research. *Communications of the ACM*, 45(10):109–111, 2002.

[5] B.N. Schilit, M.N. Price, G. Golovshinski, K. Tanaks, and C.C. Marshall. The reading appliance revolution. *IEEE Computer*, 32(1):65–73, 1999.

[6] James F. Voss and Laurie Ney Silfies. Learning from history texts: The interaction of knowledge and comprehension skill with text structure. *Cognition and Instruction*, 14(1):45–68, 1996.

[3]Micro- and macro-average are identical, since the same number of answers is collected for each test item.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Verification of Proof Steps for Tutoring Mathematical Proofs[1]

Dominik DIETRICH [a] and Mark BUCKLEY [b]
[a] *Dept. of Computer Science*
[b] *Dept. of Computational Linguistics*
Saarland University, Postfach 151150, 66041 Saarbrücken, Germany
dietrich@ags.uni-sb.de *buckley@coli.uni-sb.de*

1. Motivation

The feedback given by human tutors is strongly based on their evaluation of the correctness of student's contributions [7]. A typical approach to verification of contributions in ITSs is model tracing, in which contributions are matched against a precomputed solution graph [1,3]. However verifying steps in mathematical theorem proving poses particular problems because there are possibly infinitely many acceptable proofs, steps may be only partially ordered, and the student should be free to build any correct solution. Tracing contributions against static solution graphs is therefore overly restrictive for exercises in typically sized mathematical theories.

We propose a flexible domain-independent method of verifying proof steps of varying lengths on-the-fly for a mathematics tutoring system. Our approach can verify steps in unforeseen proofs and incrementally builds a solution state for each possible interpretation of underspecified or ambiguous steps. We utilise an existing mathematical reasoner ΩMEGA [8] and existing mathematical knowledge. Our approach to the verification of proof steps is motivated by our corpus of Wizard-of-Oz tutorial dialogues between students and experienced mathematics teachers [4]. In a proof of the theorem $(R \cup S) \circ T = (R \circ T) \cup (S \circ T)$, the student's first proof step "Let $(x, y) \in (R \cup S) \circ T$" consists of two definitions: set extensionality and subset. The tutor responds with "Correct!". This shows the tutor must know the correctness of steps to apply pedagogical strategies and that proof steps can contain applications of many mathematical definitions.

2. The ΩMEGA Prover

The development of the proof assistant system ΩMEGA is one of the major attempts to build an all-encompassing assistance tool for the working mathematician or for the formal work of a software engineer. It is a representative of systems in the paradigm of *proof planning* and combines interactive and automated proof construction for domains with rich and well-structured mathematical knowledge.

[1]Supported by the Collaborative Research Centre on *Resource-Adaptive Cognitive Processes*, SFB 378.

In ΩMEGA, proofs are represented at the *tasklayer*, which is an instance of the proof data structure (PDS) [2]. Each proof attempt is represented by an agenda, which maintains a set of subproblems to be solved to finish the proof of the overall conjecture. The nodes of the PDS are annotated with *tasks*, which are Gentzen-style multi-conclusion sequents augmented by a means to define multiple foci of attention on subformulas. Proof search is realised by reducing a task to a possibly empty set of subtasks by transformation rules which can be applied to subformulas. The basic transformation rules are *inferences*, representing the application of theorems, definitions, and axioms.

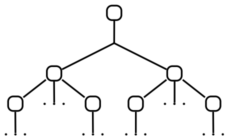
3. Representing and Updating the Student's Cognitive Proof State

We now show how the ΩMEGA-prover can be used to represent possible cognitive states the student might be in, and to judge about the correctness of a proof step of the student.

Representing the Possible Cognitive Proof States The PDS is used to simultaneously represent all of the possible cognitive proof states the student might be in, where each agenda corresponds to one possible cognitive proof state. In addition, the subproblem the student is currently working on is marked. Given a problem and a theory, an *initial PDS* is constructed, containing the problem the student has to solve. The *initial mental state* is represented by the agenda containing the problem.

Updating the Cognitive Proof States Given a set of possible cognitive proof states and a preprocessed utterance, we must determine all possible successor states which are consistent with the utterance.For the sake of simplicity we only show how to determine the successor states of a single cognitive proof state. Combining the result for each state gives the complete set of successor states. If no agenda can provide a consistent successor state then we conclude that the step is incorrect.

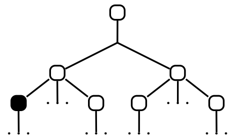
Given a classified utterance its successor agendas are determined in four steps: (i) Performing a depth limited BFS search, resulting in a set of successor nodes, (ii) selecting consistent successor nodes and creating partial agendas out of them, (iii) completing the partial agendas, (iv) cleaning the PDS. The overall result is a confirmation of whether the step could be verified, with the side-effect that the PDS has been updated to contain exactly the possible cognitive proof states resulting from the performance of the step.



Step (i) The node representing the problem the student is currently working on is expanded using a depth limited BFS search. The depth limiter specifies how many calculus steps the student is allowed to perform implicitly. Intuitively we can think of this bound as reflecting the experience of the student, and should be based on a student model. This bound is needed to guarantee termination of the verification algorithm, which might otherwise not terminate for a faulty step even if we assume a complete calculus. The correspondence of student steps to calculus steps may vary for each calculus. Tests show that 3 is a good choice for our domain.

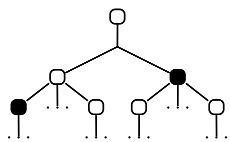
In our example the initial agenda contains the task $\vdash (R \cup S) \circ T = (R \circ T) \cup (S \circ T)$. The search tree is shown schematically on the right. To verify the student's proof step the node containing this task is expanded using the axioms from the theory, whereby each task is successively reduced to a list of subtasks.

Step (ii) We suppose that for each class of proof step there is a filter which chooses exactly those successor states consistent with the student's step. For example, a task is consistent with respect to a step "let$f$" if it contains the (sub)formula f as an assumption, and the free variables of f have been introduced as new eigenvariables. To further restrict the consistent successor nodes only those with minimal distance to the expansion node are stored.

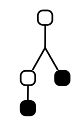


In the example the filter returns exactly one consistent successor node at depth 2, namely the node associated with the task $(x, y) \in (R \cup S) \circ T \vdash (x, y) \in (R \circ T) \cup (S \circ T)$. Indeed, this task justifies the introduction of $(x, y) \in (R \cup S) \circ T$. A new partial agenda is created in which the task is the new selected subgoal.

Step (iii) The agendas obtained from step (ii) can be partial, i.e. not all subgoals that must be solved are in the agenda. Such situations occur if the prover reduces a task to multiple subtasks but the filter does not select a node from each branch. In this case the user has not modified any of the missing subtasks. Hence it is reasonable to extend the partial agenda by the tasks introduced by the reduction. In the example above we extend the agenda as shown on the right.



Step (iv) Usually there will be many nodes which are generated by the search but are rejected by the filter. All those nodes are removed from the PDS. In our example this results in the PDS as shown. The new cognitive state of the student could be uniquely identified and thus the proof step was correct.



Conclusion and Related Work Using proof search to verify steps within the current proof situation removes the need for precomputed solution graphs, and our approach is thus more flexible than approaches which check contributions by matching them to a precomputed solution graph [3,5,6,9]. It has been implemented in the ΩMEGA system. Future work will include investigating the filters which are applied to proof states and verifying the goal-directedness of steps.

References

[1] V. Aleven, K. Koedinger, and K. Cross. Tutoring Answer Explanation Fosters Learning with Understanding. In *Proc. of AIED*, pages 199–206, Amsterdam, 1999. IOS.

[2] S. Autexier, C. Benzmüller, D. Dietrich, A. Meier, and C.-P. Wirth. A generic modular data structure for proof attempts alternating on ideas and granularity. In *Proc. of MKM*, Bremen, 2006. Springer.

[3] Cristina Conati, Abigail Gertner, and Kurt Vanlehn. Using Bayesian Networks to Manage Uncertainty in Student Modeling. *User Modeling and User-Adapted Interaction*, 12(4):371–417, 2002.

[4] C. Benzmüller *et al.* A corpus of tutorial dialogs on theorem proving; the influence of the presentation of the study-material. In *Proc. of LREC*, Genoa, 2006. ELDA.

[5] Arthur C. Graesser, Katja Wiemer-Hastings, Peter Wiemer-Hastings, and Roger Kreuz. Autotutor: A simulation of a human tutor. *Cognitive Systems Research*, 1:35–51, 1999.

[6] N. Heffernan and K. Koedinger. Building a 3rd generation ITS for symbolization: Adding a Tutorial Model with Multiple Tutorial Strategies. In *Proc. of the ITS Workshop in Algebra Learning*, 2000.

[7] Gregory D. Hume, Joel A. Michael, Rovick A. Allen, and Martha W. Evens. Hinting as a tactic in one-on-one tutoring. *Journal of the Learning Sciences*, 5(1):23–47, 1996.

[8] Jörg Siekmann, Christoph Benzmüller, and Serge Autexier. Computer Supported Mathematics with ΩMEGA. *Journal of Applied Logic*, 4(4):533–559, 2006.

[9] C. Zinn. Supporting Tutorial Feedback to Student Help Requests and Errors in Symbolic Differentiation. In M. Ikeda, K. D. Ashley, and T.-W. Chan, editors, *Proc. of ITS*, pages 349–359. Springer, 2006.

Motivationally Intelligent Systems: Diagnosis and Feedback

Benedict DU BOULAY [a], Genaro REBOLLEDO MÉNDEZ [a], Rosemary LUCKIN [b] and Erika MARTÍNEZ-MIRÓN [c]
[a]*Human Centred Technology Group, University of Sussex, Brighton, BN1 9QJ UK*
[b]*The London Knowledge Lab, University of London, London WC1N 3QS UK*
[c]*Centro de Ciencias Aplicadas y Desarrollo Tecnológico, Universidad Nacional Autónoma de México, México 04510*

Abstract: Motivationally intelligent systems deploy resources and tactics dynamically to *maintain or increase* the student's *desire to learn* and her *willingness to expend effort* in so doing. Three categories of diagnostic inputs and feedback reactions are outlined each with its associated meta-level. The meta-level includes the account which learners tell themselves, the system and others about what they know, how they feel, and the conditions under which they learn best.

Keywords: motivation, affect, evaluation of meta-cognitive and affective issues

Introduction

Motivational pedagogy that can be applied in an AIED context comprises two distinct kinds of theory: (i) how systems should act or react in order to change the motivational state of the student, and (ii) how a student's motivational state affects her learning.

1. Motivational diagnosis and feedback

In terms of the first kind of theory, [1,2,3] provide qualitative guidance. In terms of process models, at one end of the continuum there are complex models of general emotional processing that model how emotions emerge, develop and change. OCC is a well known instance [4]. At the other end of the continuum are "thermostat" models of motivation, based on interactions between a small number of motivational variables (e.g. [5]). Finally there are models somewhere in between that cover only those emotions that are relevant in educational situations, motivation included (e.g. [6]).

As far as how motivation affects learning, there are various accounts of the complex relations between motivation and other issues, such as goal-orientation, metacognition, values and beliefs, and (indeed) emotion (e.g. [7, 8]). Given the complexity of operationalising educational theory in systems, new useful approaches are being adopted to mine the large amounts of user interaction data now available. These methods are used to find relationships within that data and to measurable variables such as post-test performance (e.g. via hidden Markov models, [9]).

We define three broad categories within which motivationally intelligent systems operate together with their associated diagnostic inputs and feedback reactions, see

Table 1. By “diagnostic input” we mean the kind of event or measurement that provides input data to the system, such as the student asking for help, dominating a discussion, or their posture. By “feedback reactions” we mean actions or outputs by the system, such as changing the facial expression of an online agent, setting a harder problem, putting two students in touch with each other and so on. Note that each of the categories has a “meta-level”. This corresponds to the degree that the student (and the system) is able to reflect on and articulate the impact of that level on her learning.

Table 1: Categories of diagnostic input and feedback reaction.

CATEGORY		DIAGNOSTIC INPUTS	FEEDBACK REACTIONS
DOMAIN	Knowledge and skills of the student.	Performance, latencies, effort, focus of attention [10]	Activity choice, pace or order of work, provision of help [5]
META-COGNIT-IVE	What the student knows, can articulate and regulate about her knowledge and skills	Difficulty of work chosen, use of available help (including gaming), goal orientation [11]	Conversation about performance, degree of challenge, use of help, narrative framework [12]
AFFECT-IVE	How the student feels about the learning activity	Demeanour of student e.g. happy, engaged [13]	Praise, encouragement, criticism, politeness, teacher’s demeanour [14]
META-AFFECT-IVE	What the student knows, can articulate and regulate about her actual and expected feelings	Comments from student about expectations of feelings, motivation [15]	Conversations about expectations of feelings, state of motivation, engagement [16]
PHYSIO-LOGICAL	Bodily aspects such as heart and breathing rate, skin conductance, facial expression, body language and posture.	Sensors: skin, body movements, Cameras: facial expression, posture [3]	Breathing exercises, mantras, pauses [17]
META-PHYSIO-LOGICAL	What the student knows, can articulate and regulate about her physiological responses.	Comments from student about her body	Conversations about physiological response
CONTEXT	The spatial, social and temporal milieu within which the student is learning.	Location e.g. classroom, home, library, why learning [18]	Use of available peers and others, change of location, lighting [19]
META-CONTEXT	What the student knows, can articulate and regulate about the learning context.	Comments from the student about the context	Conversations about the nature of the context

2. Conclusions

Three categories of system input and output have been identified, each with an associated meta-level. Just as a reflectively self aware student will be able to reason about how each category bears on her learning, so a challenge for the design of motivationally intelligent systems is to reason in a similar fashion and also converse with the student at that level. Few systems have attempted to interact with the learner at the meta-levels: for example, in the case of the meta-affective, discussing with the learner the kinds of feelings that they are likely to experience in future learning

interactions or inviting self-reflection from learners about how past learning experiences felt. Similar arguments can be made for meta-physiological and meta-context discussions: “why can’t I concentrate?”, “why do I need music on to work?”.

References

[1] Lepper, M. R., Aspinwall, L. G., Mumme, D. L., and R.W. Chabay (1990). Self-perception and social-perception processes in tutoring: Subtle social control strategies of expert tutors. In J. M. Olson & M. P. Zanna (Eds.), *Self-Inference Processes: The Ontario Symposium*, Vol. 6 (pp. 217-237). Hillsdale, NJ: Lawrence Erlbaum Associates.

[2] Keller, J. M. (1983). Motivational Design of Instruction. In *Instructional-Design theories and models: An overview of their current status*. C. M. Reigeluth (Ed.). Hillsdale, Erlbaum. pp 383-434.

[3] Prendinger, H., Mori, J. and M. Ishizuka (2005) Recognizing, modelling, and responding to users' affective states in Liliana Ardissimo, Paul Brna, and Antonija Mitrovic, editors, *Proceedings of 10th International Conference, User Modelling 2005, Edinburgh*, number 3538 in Lecture Notes in Artificial Intelligence, pp 60-69. Springer-Verlag.

[4] Orthony, A., Clore, G. L., & Collins, A. (1988). *The cognitive structure of emotions*. Cambridge University Press, Cambridge, New York, New Rochelle, Melbourne, Sydney.

[5] del Soldato, T. and B. du Boulay (1995). Implementation of motivational tactics in tutoring systems. *International Journal of Artificial Intelligence in Education* **6**: 337-378.

[6] Conati C and Mclaren H. (2005). Data-driven Refinement of a Probabilistic Model of User Affect. *Proceedings of UM2005 User Modeling: Proceedings of the Tenth International Conference,* Lecture Notes in Computer Science, Volume 3538/2005, Springer Berlin / Heidelberg

[7] Pintrich, P. R. and E. V. d. Groot (1990). "Motivation and self - regulated learning components of classroom academic performance" *Journal of Educational Psychology* **82**(1): pp 33-40.

[8] Pintrich, P. R. (2000). "An achievement goal perspective on issues in motivation terminology, theory, and research" *Contemporary Educational Psychology* **25**(1): 92-104.

[9] Soller, A. and A. Lesgold (2003) A computational approach to analyzing online knowledge sharing interaction. In U. Hoppe, F. Verdejo and J. Kay (Eds) Artificial *Intelligence in Education: Shaping the future of learning through intelligent technologies.* pp 253-268, IOS press.

[10] Qu, L. and W. L. Johnson (2005). Detecting the Learner's Motivational States in An Interactive Learning Environment. In *Artificial Intelligence in Education: Supporting Learning through Intelligent and Socially Informed Technology*, pp 547-554, IOS Press.

[11] Heiner, C. , Beck, J. and J. Mostow (2005) When do students interrupt help? Effects of individual differences. Workshop on Motivation and Affect in Educational Software, in *AIED2005, 12th International Conference on Artificial Intelligence in Education*, Amsterdam, pp 47-49, IOS Press.

[12] Luckin, R. and L. Hammerton (2002). Getting to know me: helping learners understand their own learning needs through Metacognitive scaffolding. *Proceedings of the sixth Conference on Intelligent Tutoring Systems*. Biarritz, France, Berlin : Springer.

[13] Arroyo, I. and B. Woolf (2005) Inferring learning and attitudes from a Bayesian Network of log file data. In *Artificial Intelligence in Education: Supporting Learning through Intelligent and Socially Informed Technology*, in Frontiers in Artificial Intelligence and Applications, pages 33-40. IOS Press.

[14] Blanchard, E, & C. Frasson (2004) An autonomy-oriented system design for enhancement of learner's motivation in e-learning. In J. C. Lester, R. M.Vicari, F. Paraguacu *Intelligent Tutoring Systems: 7th International Conference, ITS 2004*, Maceio, Alagoas, Brazil, Springer Lecture Notes in Computer Science 3220, pp 34-44, Springer.

[15] Beal, C.R. and H. Lee (2005) Creating a pedagogical model that uses student self reports of motivation and mood to adapt ITS instruction. Workshop on Motivation and Affect in Educational Software, at AIED2005, 12th International Conference on Artificial Intelligence in Education, Amsterdam, pp 39-46.

[16] Eccles, J. S. and A. Wigfield (2002). "Motivational beliefs, values, and goals" *Annual Review Psychology* **53**: 109-132.

[17] Yusoff, M.Z. and B. du Boulay (2005) Integrating domian-independent strategies into an emotionally sound affective framework for an intelligent tutoring system. In *Proceedings of the Symposium on Agents that Want and Like: Motivational and Emotional Roots of Cognition and Action, at AISB'05:,* University of Hertfordshire, pages 114-117.

[18] Wolters, C.A. and P.R. Pintrich (1998) Contextual differences in student motivation and self-regulated learning in mathematics, English, and social studies classrooms. *Instructional Science*, **26**, 27-47.

[19] Johnson, W. L., J. Rickel, et al. (2000). "Animated Pedagogical Agents: Face-to-Face Interaction in Interactive Learning Environments" *International Journal of Artificial Intelligence in Education* **11**: pp 47-78.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Choosing Reading Passages for Vocabulary Learning by Topic to Increase Intrinsic Motivation

Michael HEILMAN [a], Alan JUFFS [b] and Maxine ESKENAZI [a]
[a] *Language Technologies Institute, Carnegie Mellon University*
5000 Forbes Avenue, Pittsburgh, PA 15221 USA
[b] *Department of Linguistics, University of Pittsburgh*
Pittsburgh, PA 15260 USA

Abstract: Intrinsic motivation has been shown in previous research to lead to better learning. In order to increase intrinsic motivation, REAP, a tutoring system for ESL vocabulary was enhanced to prefer practice readings that match personal interests. In a randomized experiment, students receiving personalized readings indicated higher levels of interest in post-reading questionnaires. Additionally, overall post-test scores were higher (but not significantly) for students with interest-matched practice readings than for students using a previous version of REAP that did not match topics to student interests.

Introduction

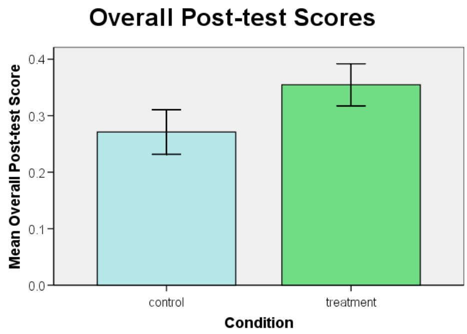
This paper discusses the enhancement of the REAP tutor [1] to allow for personalization of reading materials by topic in order to increase learner interest and intrinsic motivation. In this work, the term “personalization” refers to the selection of practice readings in order to match a student’s interests. The REAP tutor is an intelligent tutoring system for English as a Second Language (ESL) vocabulary and reading practice. It provides contextualized practice on individualized vocabulary lists by selecting reading passages (roughly 1000 words long) from a large corpus of annotated Web documents. There are a variety of constraints that the tutor considers when selecting readings for students, including reading difficulty level, grammaticality, scheduling of practice, the length of a reading, the number of target words in a reading, etc. The tutor selects reading materials from its corpus that contain target words from individualized lists and satisfy these other constraints.

Students work through a series of readings, each of which is followed by practice exercises for the target words in the reading. While reading a passage, students are able to access dictionary definitions for any word in a reading either by clicking on a highlighted target word or by typing a word into a box in the lower-left corner of the screen. The target words in the readings are also highlighted to encourage the coordination of multiple sources of information about a word’s meaning—namely, the implicit context around words and the explicit definitions of words.

Table 1: Post-reading interest responses for students using versions of tutor with or without personalization of readings.

Interest Rating	% of Scores (No Personalization)	% of Scores (Personalization)
1 (least)	6.9	4.1
2	15.0	19.1
3	**41.9**	32.0
4	30.0	**39.7**
5 (most)	6.2	5.2

Figure 1: Overall post-test scores by condition. Error bars indicate standard error. Maximum score is 1.0.



1. Text Classification for Personalization of Reading Material

To allow for the personalization of readings, the REAP tutor includes personalization by topic as a factor in its algorithm for choosing optimal readings. Students take a short survey to inform the system about which general topics they are interested in reading about. The system then prefers readings that have been classified as pertaining to those topics.

In order to identify texts that match up with student interests, a text classification system was implemented to classify each potential reading by its general topic. A Support Vector Machine [2] text classifier with a linear kernel was trained on Web pages from the Open Directory Project (ODP, http://dmoz.org), which are organized into a hierarchy of topics. SVM-Light [3] was used as the implementation of the Support Vector classifier. The following general topics were manually selected from the set of top-level ODP categories: Arts, Business, Computers, Games, Health, Home, Recreation, Science, Society, and Sports. Web pages with human-assigned topic labels from the ODP (1,000 pages/topic) were used as training data for the classifier.

Post-reading interest questionnaire results indicate that the topic choice system in REAP is effective at improving interest. After each reading, students were asked how interesting the just-completed reading was on a Likert scale from one to five, with five indicating greatest interest. Students in the treatment condition (described below) with personalization of readings by topic responded that they were interested in readings more frequently than did students in the control condition. The distribution of responses is shown in Table 1.

Personalization does not, however, mean that students learn only narrow-coverage words that relate to their topics of interest. Students with different interests practiced similar sets of general-purpose vocabulary from the Academic Word List [4]. For instance, in the study described in this paper, one student interested in arts saw the word “endure” in a text describing an artist’s early career struggles (“For an artist who has *endured* so many years of obscurity...”). Another student interested in business saw the same word used to describe economic hardship ("As California has *endured* a burst tech bubble, costly energy crisis and a staggering burden on its business community...").

2. Experimental Evaluation of Learning Gains

An experiment was conducted to measure the effects of personalization on learning progress in the REAP tutor. Thirty-five students at the English Language Institute at the University of Pittsburgh participated in this experiment as part of an intermediate English as a Second Language Reading course. The students were randomly assigned to control or treatment conditions. For students in the control condition, the REAP tutor ignored the student interest survey and offered readings to students based on the goals of the curriculum. For students in the treatment condition, the REAP tutor was the same as in the control except that it also preferred readings about topics of personal interest.

At the end of the series of 9 forty-minute training sessions, students took a post-test consisting of cloze questions for the target vocabulary words that were identified as unknown through a self-assessment pre-test. The post-test consisted of forty questions for target words that appeared in at least one passage completed by the student.

The effect of personalization on learning was measured by student performance on the post-test cloze questions for target vocabulary words. Students in the treatment condition performed better on average (M=35.5%, SD=14.9%) in terms of overall post-test scores compared to students in the control condition (M=27.1%, SD=17.2%), as shown in Figure 1. The difference in mean overall post-test scores in the treatment condition was 8.4% (95% CI = -2.8%, 19.5%), which corresponds to a medium effect size of 0.51. However, this difference is not statistically significant ($p=0.14$, two-tailed t-test).

The findings of this work suggest that automatic techniques can be effectively applied to select readings by topic to match student interests. The results for measures of learning are promising and suggest that the effects of personalization of texts for vocabulary practice should be investigated further.

Acknowledgements

The authors would like to acknowledge Kevyn Collins-Thompson, Jamie Callan, and Lois Wilson for their work on the REAP project. Thanks also go to the anonymous reviewers. This material is based on work supported by NSF grant IIS-0096139. Any opinions, findings, conclusions or recommendations expressed in this material are the authors', and do not necessarily reflect those of the sponsor.

References

[1] Heilman, M., Collins-Thompson, K., Callan, J. & Eskanazi, M. (2006). Classroom success of an Intelligent Tutoring System for lexical practice and reading comprehension. *Proceedings of the Ninth International Conference on Spoken Language Processing.*

[2] Vapnik, V. N. (1995), *The Nature of Statistical Learning Theory*. Springer.

[3] Joachims, T. (1999). Making large-Scale SVM Learning Practical. Advances in *Kernel Methods – Support Vector Learning*, B. Schölkopf and C. Burges and A. Smola (ed.), MIT-Press.

[4] Coxhead, A. (2000). A New Academic Word List. *TESOL Quarterly*, 34(2): 213-238.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Emotions and Learning with AutoTutor

Arthur GRAESSER, Patrick CHIPMAN, Brandon KING,
Bethany McDANIEL, and Sidney D'MELLO
Institute for Intelligent Systems, The University of Memphis
365 Innovation Drive, University of Memphis, Memphis, TN, 38152, USA

Abstract. The relationship between emotions and learning was investigated by tracking the emotions that college students experienced while learning about computer literacy with AutoTutor. AutoTutor is an animated pedagogical agent that holds a conversation in natural language, with spoken contributions by the learner. Thirty students completed a multiple-choice pre-test, a 35-minute training session, and a multiple-choice post-test. The students reviewed the tutorial interaction and were stopped at strategically sampled points for emotion judgments. They judged what emotions they experienced on the basis of the dialogue history and their facial expressions. The emotions they judged were boredom, flow (engagement), frustration, confusion, delight, surprise, and neutral. A multiple regression analysis revealed that post-test scores were significantly predicted by pre-test scores and confusion, but not by any of the other emotions.

Keywords: Tutorial dialogue, emotions, learner modeling

Introduction

A satisfactory understanding of the connections between emotions and complex learning is necessary to design engaging learning environments that motivate students to learn. In order to systematically investigate these relationships, we are in the process of developing a version of AutoTutor that is sensitive to both the cognitive and affective states of the learner [1, 2]. Such an affect-sensitive tutor would presumably enhance the intelligent learning environment [1, 2, 3].

AutoTutor is an intelligent tutoring system that helps students learn by holding a conversation in natural language [4]. An automated emotion classifier is necessary for AutoTutor to be responsive to learner emotions. We have previously reported some studies that collect the dialogue history, facial action units, position of their body, and other sensory channels while they learn and emote aloud [1, 5]. The features from the various modalities can be detected in real time automatically on computers, so we are currently integrating these technologies with AutoTutor.

The present study investigated the relationship between learning and the emotions that college students experience while interacting with AutoTutor on the topic of computer literacy. Whereas our previous studies of AutoTutor had learners enter their contributions through keyboard, this is the first AutoTutor study that had students make their contributions through speech. Properties of speech should be diagnostic of the learners' emotions [6].

1. Methods

The participants were 30 undergraduates at the University of Memphis who participated for extra course credit. The experiment consisted of a pre-test, an interaction with AutoTutor, a post-test, and judgments of emotions the learner experienced during the session with AutoTutor. The participants were tutored with AutoTutor on one of three major computer literacy topics: hardware, operating systems, or the internet. The pre-test and post-tests consisted of multiple choice questions that had been used in previous research on AutoTutor [7]. The 10 questions on each test tapped deep levels of reasoning, causality, and explanations. Performance in these tests was simply the proportion of questions answered correctly.

Participants interacted with AutoTutor for 35 minutes on one of the three randomly assigned topics in computer literacy. A detailed discussion of the architecture, strategies, and effectiveness of AutoTutor are provided in previous publications, as cited above. Each major topic had 6 tutoring questions that required about a paragraph of information in a good answer. We used the commercially available Dragon Naturally Speaking™ (v 6) speech recognition system for speech-to-text translation.

Students viewed their own session with AutoTutor after interacting with the tutor and completing the post-test. The judgments for a learner's tutoring session proceeded by playing a video of the learner's face along with the dialogue history. The students were instructed to make judgments on what affective states were present at three different points during the tutorial dialogue: (1) immediately after AutoTutor gave the short feedback (positive, neutral, negative) during a turn, (2) immediately before the learner started expressing his/her spoken turn, and (3) other randomly selected points in the dialogue. The students also had the option of going back in between these points and making emotion judgments. The data collection program provided a checklist of emotions for them to mark at these points. The participant was instructed to mark the affect state that was most pronounced at each point.

A list of the affective states and definitions was provided for the learners. The states were boredom, confusion, flow, frustration, delight, surprise, and neutral. These were the affective states that were most frequently experienced in previous studies of AutoTutor [5, 8, 9] that investigated emotions with alternative methods: Emote-aloud procedures during learning and observations of trained judges or peers.

2. Results and Discussion

A number of scores were computed for each of the 30 college students. These included pre-test scores, post-test scores, and the proportion of first-choice emotion judgments in each of the 7 emotion categories. The test scores varied from 0 to 1.

The post-test scores were significantly predicted by confusion (r = .490), but not the other emotion categories. We performed a multiple regression analysis that assessed the extent to which post-test scores were predicted by pre-test scores and confusion. The regression equation was significant, $F(2, 27) = 6.50$, $p < .05$, $R^2 = .326$. Confusion was a significant predictor ($p < .05$, two-tailed test, $\beta = .421$), as was also the pre-test scores ($p < .05$, one-tailed test, $\beta = .300$). This significant effect of confusion replicates a previous study that had trained judges observe student-AutoTutor interactions and record emotions every 5 minutes [8]. Confusion is a signal that the learner is experiencing cognitive disequilibrium and thinking.

This is yet another study that substantiates the importance of confusion (perplexity) in complex learning [8, 9]. When the learner is confused, they are in the state of cognitive disequilibrium, heightened physiological arousal, and more intense thought. In contrast, post-test scores were not significantly predicted by the other emotions (flow, boredom, frustration, delight, surprise) that were retrospectively identified by the learners when they viewed a recording of their learning experience. These other emotions presumably play a more prominent role in other learning environments and other populations of learners.

Our next step is to build an emotion-sensitive AutoTutor that will promote both learning gains and more engagement in the learner. AutoTutor should have different strategies and dialogue moves when the learner is confused, frustrated, bored, versus experiencing flow – the four most frequent emotions we have found in our work with AutoTutor. Since confusion is tightly linked to learning, it will be important to have the dialogue moves manage the learner's confusion productively. AutoTutor's mechanisms will need to be sensitive to cognitive and motivational characteristics of the learners in addition to their emotional state. We have already designed a computational architecture and algorithms to automatically sense the four major emotions (confusion, frustration, boredom, and flow) on the basis of the dialogue history, facial expressions, and body posture [1]. Whether this new emotion-sensitive AutoTutor will help learning awaits future research.

Acknowledgements

This research was supported by the National Science Foundation (ITR 0325428, REC 0633918). Any opinions, findings, conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of NSF.

References

[1] S. K. D'Mello, R. Picard and A. C. Graesser, Toward an affect-sensitive AutoTutor, *IEEE Intelligent Systems* (in press).

[2] A. C. Graesser, G. T. Jackson and B. McDaniel, AutoTutor holds conversations with learners that are responsive to their cognitive and emotional states, *Educational Technology* **47** (2007), 19-22.

[3] M. R. Lepper and J. Henderlong, Turning "play" into "work" and "work" into "play": 25 years of research on intrinsic versus extrinsic motivation, *Intrinsic and extrinsic motivation: The search for optimal motivation and performance* (2000), 257-307.

[4] A. C. Graesser, P. Chipman, B. C. Haynes, and A. Olney, AutoTutor: An intelligent tutoring system with mixed-initiative dialogue, *IEEE Transactions on Education* **48** (2005), 612-618.

[5] S. K. D'Mello, S. D. Craig, and A. C. Graesser, Predicting affective states through an emote-aloud procedure from AutoTutor's mixed-initiative dialogue, *International Journal of Artificial Intelligence in Education* **16** (2006), 3-28.

[6] D. J. Litman and K. Forbes-Riley, Predicting student emotions in computer-human tutoring dialogues, *Proceedings of the 42nd annual meeting of the association for computational linguistics* (2004), 352-359.

[7] A. C. Graesser, S. Lu, G. T. Jackson, H. Mitchell, M. Ventura, A. Olney, and M. M. Louwerse, AutoTutor: A tutor with dialogue in natural language, *Behavioral Research Methods, Instruments, and Computers* **36** (2004), 180-193.

[8] S. D. Craig, A. C. Graesser, J. Sullins, and B. Gholson, Affect and learning: An exploratory look into the role of affect in learning with AutoTutor, *Journal of Educational Media* **29** (2004), 241-250.

[9] A. C. Graesser, B. McDaniel, P. Chipman, A. Witherspoon, S. K. D'Mello, and B. Gholson, Detection of emotions during learning with AutoTutor, *Proceedings of the 28th Annual Meetings of the Cognitive Science Society* (2006), 285-290.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Standard-Compliant Scenario Building with Theoretical Justification in a Theory-Aware Authoring Tool

Yusuke HAYASHI[a], Jacqueline BOURDEAU[b], Riichiro MIZOGUCHI[a]
[a] *ISIR, Osaka University,* [b] *LICEF, Télé-université*
[a] *8-1 Mihogaoka, Ibaraki, Osaka, 567-0047 Japan*
[b] *100 Sherbrooke W., Montréal, (QC) H2X 3P2 Canada*

Abstract. Nowadays standard technologies play important roles in enhancing sharability, reusability and interoperability of learning contents. However, there is a lack of pedagogical justification of the contents implemented with the standards. This paper discusses the standard-compliance of our ontology-based modeling framework and how the framework gives theoretical justification to standard-compliant learning/instructional scenarios in a theory-aware authoring tool.

Introduction

Nowadays standard technologies play important roles in enhancing sharability, reusability and interoperability of learning contents. However, it is pointed out that there is a lack of pedagogical justification of the contents implemented with the standards [5].

In this study we take an ontological engineering approach to organize educational theories in a formal and computer-understandable way [4]. Through this approach we have proposed a comprehensive ontology[1] that covers different theories and paradigms, and have developed a modeling framework of learning/instructional scenarios based on the ontology [1][2]. This paper discusses the standard-compliance of our modeling framework and how the framework gives theoretical justification to standard-compliant learning/instructional scenarios in a theory-aware authoring tool.

1. Standard-compliant scenario building on an ontological modeling framework

In our framework, a scenario can be modeled as a hierarchical structure of "Instructional_Leaning (I_L) event", which are composed of instructional and learning actions for achieving a certain change of a learner state [1]. We call the model an "I_L event decomposition tree". The basic idea of the model is to relate a macro-I_L event to the lower (micro) ones that collectively achieve the upper (macro) I_L event in terms of a learner state (The relation is referred to as "WAY" in this study). Currently, we have organized about 100 pieces of WAY based on some theories [2]. Such WAYs are called WAY-knowledge.

[1] The ontology is opened to the public on our OMUNIBUS project web page (http://edont.qee.jp/omnibus/).

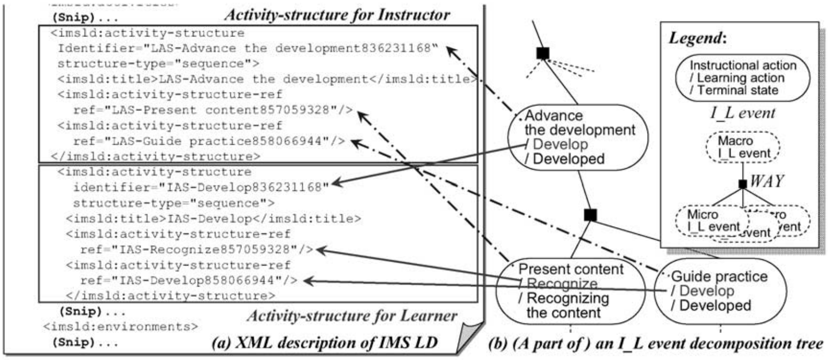


Fig. 1 Mapping an I_L event decomposition tree into IMS LD

In order to enhance sharability and reusability of the scenario descriptions, we have mapped I_L event decomposition tree onto IMS LD specifications [3]. Briefly speaking, each unit of decomposition in an I_L event decomposition tree can be converted to two `activity-structures` for `learner` and `instructor` in an IMS LD description as shown in Fig. 1.

In IMS LD, only top and leaf activities have the description of the objective while the others do not have. Therefore only a part of the design intention can be converted to the IMS LD description although it keeps sharability and executability of learning/instructional scenarios. On the other hand, an I_L event decomposition tree keeps the whole design intention together with theoretical justification of it. For these reasons, IMS LD and our modeling approach are complementary to each other.

2. Generation mechanism of theoretical scenario explanation

This study aims at building a theory-aware authoring tool based on our comprehensive ontology and modeling framework. One of the characteristics of such an authoring tool is its ability to interpret and explain learning/instructional scenarios in terms of theories. A descriptive concept (I_L event) and a prescriptive concept (WAY-knowledge) in our comprehensive ontology enable information systems to give explanations and suggestions about scenarios described as an I_L event decomposition tree.

In order to generate scenario explanation we made message templates whose vocabulary comes from the ontology and whose structure is partly based on an I_L event decomposition tree. Table 1 summarizes the classification of the templates. If a scenario is described based on a piece of WAY-knowledge, an interpretative explanation gives a theoretical justification. On the other hand a scenario is described only in the terms defined our ontology, a suggestive explanation offers suggestions for improvements in the scenario in terms of theories.

Table 1 A classification of explanation types and cases (not exhaustive)

		Cases
Explanation types	Interpretative	Theoretical justification (Notes: *Explaining interpretation of relation among events in a scenario in terms of a theory*)
		Theory description
		Scenario comprehension
	Suggestive	Insufficiency of necessary goals (Notes: *It seems learners can not achieve the goal because necessary (sub) goal is insufficient in the scenario.*)
		Insufficiency of supplementary goals
		Excess of goals
		Disproportion in process
		Inconsistency of principle
		Unsustained state

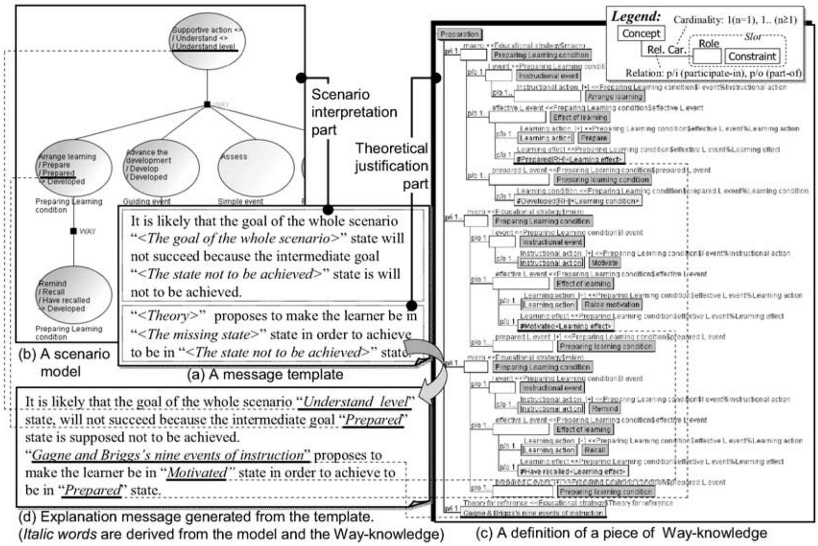


Fig. 2 An example of suggestive explanation

We are currently developing the explanation function in our prototype system named "Smarties". Fig 2 illustrates an example of suggestive explanation. Fig 2 (d) shows an example of generated messages about *Insufficiency of necessary goals*. This message is generated by the message template (Fig. 2 (a)) and the scenario interpretation based on WAY-knowledge. Italic words in the message are specified with the scenario model (Fig. 2 (b)) and a definition of a piece of WAY-knowledge (Fig. 2 (c)) in the ontology. Such a message can be generated if the scenario is described in the terms defined in the ontology.

3. Conclusion

We have discussed a functionality of theory-awareness of an ontology-based authoring tool and its compliance with standard technologies, especially focused on IMS LD specifications. Conceptual understanding of scenarios based on the theory-awareness enables an authoring tool to explain scenarios theoretically and to record them in a sharable format with theoretical justification.

References

[1] Hayashi, Y., Bourdeau, J. and Mizoguchi, R. : "Ontological Modeling Approach to Blending Theories for Instructional and Learning Design", *Proc. of ICCE2006*, pp. 37-44, 2006.
[2] Hayashi, Y., Bourdeau, J. and Mizoguchi, R. : "Ontological Support for a Theory-Eclectic Approach to Instructional and Learning Design", *Proc. of EC-TEL2006*, pp. 155-169, 2006.
[3] IMS Global Learning Consortium, Inc.: IMS Learning Design. Version 1.0 Final Specification, 2003. Retrieved Dec. 15, 2006 from `http://www.imsglobal.org/learningdesign/`
[4] Mizoguchi, R. and Bourdeau, J.: "Using Ontological Engineering to Overcome Common AI-ED Problems", *International Journal of Artificial Intelligence in Education*, Vol.11, No.2, pp. 107-121, 2000.
[5] Psyche, V., Bourdeau, J., Nkambou, R., and Mizoguchi, R.: "Making Learning Design Standards Work with an Ontology of Educational Theories", *Proc. of AIED2005*, pp. 539-546, 2005.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Using Multinets for Learner Modelling

Mathieu HIBOU
Crip5 Université René Descartes – Paris 5
45 rue des Saints-Pères 75270 Paris Cedex 06 France
mathieu.hibou@math-info.univ-paris5.fr

Jean-Marc LABAT
LIP6 Université Pierre et Marie Curie –Paris 6
France
jean-marc.labat@upmc.fr

Keywords: student modelling, Bayesian networks, multinets, cognitive diagnosis.

Abstract.
In this paper we shall introduce a new conceptual model for learner modelling based on multinets, which are Bayesian networks mixture. This conceptuel model makes it possible to take into account in a single student model different Bayesian networks, with the same nodes but different structures. We also present experiemental results obtained with real students data.

Introduction

Bayesian networks (BNs) [9] appear to be efficient tools for learner modelling and during the last decade, they have been successfully used in many systems ([2],[7],[8]). It appears that the most problematic issue concerns the choice of the arcs' orientation, or, more generally, the structuring of the BN-based student model (BNb-SM). Moreover, the question of the adaptation of the structure of the network to the student's knowledge structure remains mostly unexplored [7], even though results obtained in cognitive psychology prove the existence of structural differences between novices' and experts' knowledge [1]. The key idea of this paper is that the structure of a BNb-SM depends on the level of expertise of the student. Therefore, such a model should be constituted of several competing BNs (same nodes, different structures), instead of a single one, as it is usually done.

1. Arcs' orientation in BN-based student models

In a large majority of systems, arcs are drawn and oriented from the domain nodes to the task specific nodes. However, the question on the arcs' orientation between the nodes of the domain layer is not so straightforward. Sometimes, the arcs are drawn from the nodes representing more general knowledge to those representing more specific one, sometimes it is the contrary [6]. This choice has consequences on the independence relationships between variables, and therefore on the diagnosis obtained. In the first case, any observation on any specific node modifies the beliefs on all the other variables, whereas, in the second case, the propagation of information is far more limited.

Orientating the arcs from topic to sub-topics comes to considering that the students skills are homogeneous, whereas the opposite orientation reflects their heterogeneity. In fact, our analysis [4] has shown that this orientation seems to depend on the student's level of expertise. The more advanced in the school curriculum a student is, the more his/her skills are dependent. This is a case of asymmetric dependency [3] with which a single BN can not deal.

Therefore, we propose to model the learners with different BNs, each of them having the same nodes but different structures.

2. Multinet

A multinet is constituted of different BNs which have the same nodes, but different topologies, a probability being attached to each of these BNs [3]. If we consider a multinet with n BNs $(r_1, \ldots, r_n)$ with respective probabilities $(p_1, \ldots, p_n)$, the probability of an event E is given by the formula $P(E) = \sum_{i=1}^{n} p_i P(E|r_i)$, where $P(E|r_i)$ is the probability of E in the BN r_i. Instead of using BNb-SM, we propose to use mulinet based student model (MNb-SM).

Two different types of diagnoses, local or global, can be obtained with MNb-SM. Local diagnoses are constituted of the posterior laws of the domain nodes, given the observation of the students' interactions. They are close in their form to the diagnoses obtained with a single BN student model as they are lists of probabilities. Global diagnoses are determinations of the BNs that best fit the students' interactions, that is to say the BNs that maximise $P(r_i| \text{ student's actions})$. These diagnoses give global information about the homogeneity or heterogeneity of each student's knowledge.

3. Experiments on *Pépite* data

The *Pépite* system [6] tests mathematics competencies (14 or 15-year old pupils in forth or fifth form). It is constituted of a questionnaire and a diagnosis module that analyses the students' answers. This system already works and we only use it as a bank of data (400 students' logs, that is to say 400 sets of answers to 34 different exercises) to test our propositions and theoretical assumptions. We conceive a MNb-SM constituted of 5 BNs (figure 1) and use it to test the existence of a correlation between the student's level of expertise and the BN that represents him/her the best. We use the form the students belong to as an indicator of their expertise level. We performed different chi-squared tests, with or without learning the parameters of the MNb-SM (table 1).

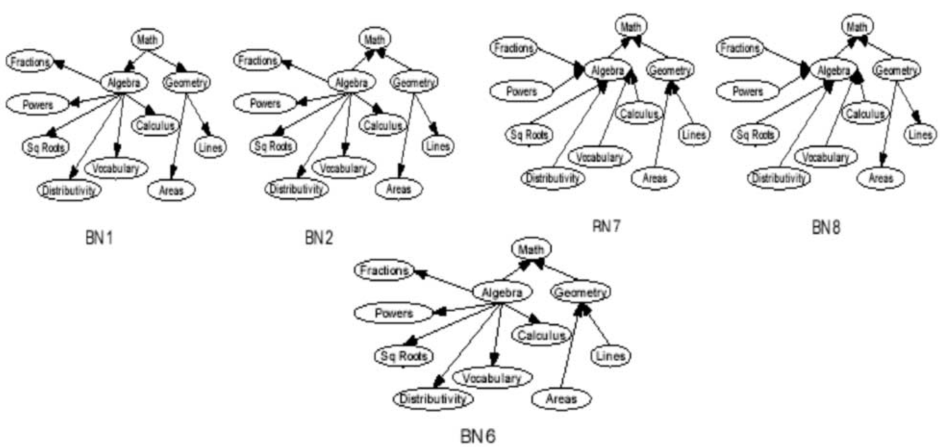


Figure 1: **The five different BNs (simplified versions) constituting the MNb-SM.**

We classified each pupil according to the result of the global diagnosis (*i.e* the BN that best fits his/her answers) (observed results in table 1). Then we compare the two forms according to this classification. The second part of each sub-tables (Form 5 or Form 4) gives

the figures that appeared if the form to which a pupil belongs and the BN that best fits his/her answers were two independent characteristics. The third part gives the respective account of each BN to the deviation from the expected figures.

In all of our experiments, the chi-squared tests clearly shows that there is a correlation between the BN that best fits the pupil’s answers and the form s/he belongs to. Moreover, the top-down structured BNs appear to be over-represented in fifth form (older pupils), and in the same way, the bottom-up structured BNs are over-represented in fourth form (younger pupils), confirming our analysis. The top-down structured BNs tend to represent pupils further in their studies, with homogeneous knowledge, while the bottom-up structured BNs suit to beginners better. We think that this is a strong evidence in favour of our position: the level of expertise and the structure of the network are not independent.

BN	Form 5				Form 4			
	Observed	Expected	$(O-E)^2/E$		Observed	Expected	$(O-E)^2/E$	
BN 8	104	118,15	1,69	-	109	94,85	2,11	+
BN 7	4	5,55	0,43	-	6	4,45	0,54	+
BN 6	25	24,41	0,01	+	19	19,59	0,02	-
BN 2	11	8,32	0,86	+	4	6,68	1,08	-
BN 1	69	56,58	2,73	+	33	45,42	3,4	-

Table 1: **comparison of the global diagnoses between the fifth and the forth form.**

Conclusion

In this paper we propose a conceptual model based on multinet to manage different BNs (same nodes, different structures) representing the same student. We give evidence (obtained with experiments made on a large experimental set (400 students logs)) in favour of the hypothesis about the correlation between the advancement of the student in the curriculum and the structure of the BN that fits his/her answers best.

Hence, we both gave evidence in favour of the need of variability of the model structure according to the evolution of the student’s knowledge and described a model that makes it possible to take into account such variations. This model is a solution to the on-line structure adaptation problem of Bayesian Networks based student models.

We strongly believe that our approach gives a new insight into the use of Bayesian Networks for student modelling, we now need to extend our research to larger size models and/or to different fields.

References

[1] Chi, M.T.H., Feltovitch, P.J., & Glaser, R. (1981). Categorization and representation of physics problems by experts and novices. *Cognitive Science, 5,* 121-152.

[2] C. Conati, A. Gertner, K. Vanlehn. Using Bayesian networks to manage uncertainty in student modeling, in *Journal of User Modeling and User-Adapted Interaction*, volume 12 (4), pages371-417, 2002.

[3] D. Geiger, D. Heckermann. Knowledge representation and inference in similarity networks and Bayesian multinets, *Artificial Intelligence*, volume 82 (1–2), pages 45–74, 1996.

[4] M. Hibou, J.-M. Labat, How to orientate arcs in a Bayesian network based student model? *Proceedings of the 6th ICALT conference*, IEEE Conference Publishing, pages 560-562, 2006.

[5] A. Jameson. Numerical uncertainty management in user and student modeling: an overview of systems and issues, in *User- Adapted Interaction*, volume 5 (3-4), n°5, pages193-251, 1996.

[6] S. Jean. PÉPITE : un système d'assistance au diagnostic de compétences. PhD Thesis, Université du Maine, 2000.

[7] M. Mayo, A. Mitrovic. Optimising ITS behaviour with Bayesian networks and decision theory, in *International Journal of Artificial Intelligence in Education*, n°12, pages124-153, 2001.

[8] R. Myslevy, D. Gitomer. The role of probability-based inference in an intelligent tutoring system, in *User-Modeling and User- Adapted Interaction*, n°5, pages 253-282, 1996.

[9] J. Pearl. *Probabilistic Reasoning in Intelligent System*, Morgan Kaufmann, 1988.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Using Ontologies for an Effective Design of Collaborative Learning Activities

Seiji ISOTANI[1] and Riichiro MIZOGUCHI
The Institute of Scientific and Industrial Research, Osaka University, Japan

Abstract. Although artificial intelligence has been successfully introduced to enhance Education through technologies in the past few years, major challenges still remain. One of them is how to represent the knowledge of intelligent systems. To represent the knowledge of systems to support collaborative learning is particularly challenging because it is based on various learning theories and given the complexity of group learning. The main objective of this work is to introduce an ontological infrastructure on which we can build well-grounded theoretical knowledge based on learning theories and to show how we can use it to develop programs to support intelligent guidance for an effective design of group activities.

Keywords. Collaborative learning, ontological engineering, intelligent educational system.

Introduction

To develop intelligent systems to support collaborative learning (CL) is especially challenging in view of knowledge representation. Current knowledge concerning CL is based on various learning theories, which are always expressed in natural language and are particularly complex given the context of group learning where the synergy among learner's interactions affect the learning processes and hence learning outcome. Without the explicit representation of learning theories it is difficult to support the design of group activities based on well-grounded theoretical knowledge.

Our approach calls upon techniques of ontological engineering to make theories "understandable" both by computers and humans. We then propose techniques to reason on these theories which contribute to dynamic guiding and instructional planning. Also we have been developing an intelligent support system that represents theories graphically to facilitate the design CL activities with theoretical justifications.

1. Representation of Learning Theories on GMIP

The use of ontological engineering for knowledge systematization has shown significant results concerning how to represent the knowledge of educational environments considering theories [2]. In CSCL (Computer Supported Collaborative Learning) research, ontologies have been successfully applied to solve problems such as: group formation, CL representation, interaction analysis and patterns and modeling

[1] Corresponding Author: Seiji Isotani, ISIR, Osaka University, 8-1 Mihogaoka, Ibaraki, Osaka, 567-0047, Japan; E-mail: isotani@acm.org

of learner's development [3]. Nevertheless, there are some limitations: (a) it is not easy to determine which learning theory is appropriate to explain and support learner's development; and (b) it is difficult to propose activities in compliant with the theories to enhance interactions among learners and lead them to achieve desired goals.

To overcome these limitations we re-analyzed seven different learning theories frequently used to support CSCL activities. And then, we proposed the ***Growth Model Improved by Interaction Patterns*** **(GMIP)** [4]. The GMIP is a graph model based on an ontological structure to describe an excerpt of learning theory. It represents, in a simplified way, the learner's knowledge acquisition process and skill development process, explaining the relationships between learning strategies, educational benefits and interactions used to achieve these benefits.

The GMIP graph has twenty nodes, which represent the levels of the learner's development at a certain moment of learning. Each node is composed by two triangles. The upper-right triangle represents the stage of knowledge acquisition, while the lower-left triangle represents the stage of skill development. The nodes are linked with arrows that show possible transitions between nodes in compliance with [1] and [5]. Using the GMIP graph, we show the benefits of learning strategies by highlighting its path on the graph and associating each arrow with interactions activities (Figure 1).

The main contributions of GMIP for CL design are (a) to allow the graphical visualization of theories and their characteristics. Thus, users can quickly interpret the theories, their benefits and propose sequence of activities in compliance with them; and (b) to provide a formal structure based on ontologies which allows systems to reason about the theories and the features (actions, roles, strategies, etc.) prescribed by them.

2. An Ontology-based Support System to Design Group Activities

To support group activities there are many learning theories (such as Anchored Instruction, Peer Tutoring, LPP, etc). Thus, to assign roles and strategies for learners in a group we can select appropriate set of learning theories considering necessary pre-conditions and desired educational benefits for learners. This flexibility of choosing different learning theories provides us with many ways to design and conduct learning processes. However, it also suggests the difficulty of selecting the appropriate set of learning theories during the instructional design to ensure learners' benefits and the consistency of learning processes. Therefore, to help users (instructors, teachers, designers, etc) to design effective group activities we need an elaborated system that considers different learning theories to support the design in compliance with them.

In order to develop a system to support the design of CL activities we have been developing **CHOCOLATO** – *a Concrete and Helpful Ontology-aware Collaborative Learning Authoring Tool*. It is an ontology-aware system that uses ontologies developed in Hozo ontology editor (http://www.hozo.jp) to provide its theoretical knowledge. One of its sub-systems called **MARI** – *Main Adaptive Representation Interface* allows to represent learning theories on the screen using the GMIP (Figure 1). Since 2006, MARI has been implemented in Java and uses Hozo API to interpret ontologies. Through the use of ontologies it allows high expressiveness and interoperability among theories and their features. It currently has 6 theories and 12 strategies, besides other information in its database. Using ontologies and the GMIP MARI can reason on the theories to select appropriate learning theories and strategies and to suggest consistent sequence of activities for learners in a group.

The **suggestions** given by our system are only guidelines for users to propose CL activities based on theories which: (a) preserves the consistency of the learning process; and (b) guarantees a suitable path to achieve desired benefits. However, expert designers can propose their own sequence of activities. In such case the system also can assist these users providing other information that can be useful in various situations.

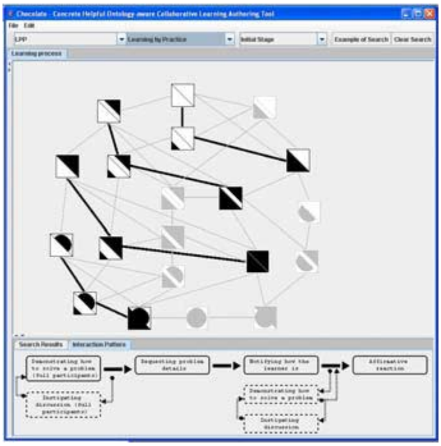


Figure 1. Screenshot of MARI showing the theory *LPP* using the strategy *Learning by Practice*. On the top it shows the path on the GMIP and on the bottom the sequence of activities in compliance with the theory.

3. Conclusions

The main contribution of this research is to introduce our model GMIP based on an ontological structure to describe learning theories for CL and to develop a support system that uses it rationally. Through the use of ontologies we aim to: (a) prevent unexpected interpretations of the theories; (b) provide a common vocabulary to describe them; (c) share and accumulate the knowledge; and (d) provide information for computational semantics which allows us to assist users based on theories.

Our approach uses a well-grounded theoretical knowledge to provide intelligent recommendations of role assignment and sequence of interactions which offers fundamental settings for an effective CL session and essential conditions to predict the impact of interactions in the learning process. Thus, we believe it is possible to provide an accumulation of knowledge about group interactions and learner's development allowing a re-formation of groups based on analysis of previous CL sessions.

References

[1] Anderson, J. R. (1982) Acquisition of Cognitive Skill, *Psychological Review*, 89(4), 369-406.

[2] Devedzic, V. (2004). Web Intelligence and Artificial Intelligence in Education. *Educational Technology & Society*, 7 (4), 29- 39.

[3] Inaba, A., Supnithi, T., Ikeda, M., Mizoguchi, R. and Toyoda, J. (2000) How Can We Form Effective Collaborative Learning Groups?. In Proc. of Intelligent Tutoring Systems, 282-291.

[4] Isotani, S. and Mizoguchi, R. (2006) A Framework for Fine-Grained Analysis and Design of Group Learning Activities. In Proc. of Int. Conference on Computers in Education, 193-200.

[5] Rumelhart, D.E., & Norman, D.A. (1978) Accretion, Tuning, and Restructuring: Modes of Learning, *Semantic factors in cognition*. LEA, 37-53.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Topic Initiative in a Simulated Peer Dialogue Agent[1]

Pamela W. JORDAN [a]
[a] *Learning Research and Development Center, University of Pittsburgh*

Abstract. One goal of our project is to create a dialogue agent that can behave as a student peer and collaborate with a human student to explain or diagnose data structures programs. A human peer must be able to initiate something the agent may not be expecting and the agent must be able to respond to it as a peer would. This paper describes the work we have completed to extend an existing natural language tutorial dialogue system to provide such a capability and the work that remains.

Keywords. linguistics and language technology, communication, collaboration

1. Introduction

One goal of our project[2] is to create a dialogue agent that can behave as a student peer and collaborate with a human student to explain or diagnose data structures programs. To participate in the negotiations that take place during peer collaboration [2], the dialogue agent must be able to both recognize and initiate a number of communicative actions, such as assertions and proposals, about the code being explained or diagnosed. Although considerable attention has been given to the difficult task of intention recognition in planning and dialogue research, relatively less has been devoted to modelling how a peer reacts to intentions that may not immediately fit with her own. Both problems are rendered more manageable by modelling either fixed-initiative interactions or mixed-initiative interactions[3] that allow only one type of initiative to be accepted for a single turn (e.g. a question from a student during system-led tutoring) or one simple, uncontested switching mechanism between system and user-led interaction (e.g. "you do the next step"). The focus of the on-going work described here is to model the much richer negotiations between peers that switch the topic initiative (e.g. from our corpus, "D: that one does need to be head = insertBeginning(88, head). C: let me just draw what it does now, and then we can figure out how to change if after").

A pure finite-state based approach is relatively easy to develop and would allow us to closely model our corpus of human-human peer interactions but does not provide enough flexibility for us to experiment with peer negotiations. Alternatively, principle-based approaches would give us the needed flexibility but would require a prohibitive amount of

[1]This work was funded by NSF grant 0536959
[2]http://andes3.lrdc.pitt.edu/PaL
[3]See [1] for a survey of theories and work on mixed-initiative.

effort to develop the domain and communication knowledge to the point we could use it in experiments with students. We chose to use the existing TuTalk dialogue system for our initial modelling effort since it provides more flexibility than pure finite-state based approaches [4]. It provides a task agenda and discourse history, it tracks discourse obligations and has been used in experiments involving large numbers of students. In addition it's modular architecture will support us in supplementing both the natural language (NL) understanding and generation modules with a human interpreter who will correct the results of both. This human assist will allow us to focus on mixed-initiative issues instead of intention recognition issues. In this paper we describe how we are extending TuTalk to provide a mixed-initiative capability for peer interactions.

2. Extending TuTalk to support topic initiative

TuTalk supports NL tutorial dialogue in which the tutor tries to elicit a main line of reasoning from the student by a series of questions similar to CIRCSIM-tutor's directed-lines of reasoning [3]. Dialogues in TuTalk can be represented as a finite state machine. Each state contains a single tutor turn. The arcs leaving the state correspond to possible student response turns. When creating a state, a dialogue author enters the text for a tutor's turn and defines several classes of student responses as transition arcs, and indicates which state each arc leads to. An arc can also "call" a finite state network, which allows authors to create hierarchical dialogues. The author creates a multi-step hierarchically-organized recipe to a cover a topic. At the leaf nodes a step comprises a state and its arcs. If no student response classes are defined then by default the transition is to the next step of the recipe. Non-terminating nodes of the recipe are a call to a subrecipe. In TuTalk a state is called an initiation and the arcs from a state its responses.

The NL associated with states and arcs is represented in concept definitions. In the simplest case, a concept is a set whose elements are NL phrases. When a string is input, the dialogue manager asks the understanding module to determine what concepts it represents and determines transitions on the basis of the concept labels returned. Likewise when a concept is to be expressed, the dialogue manager asks the generation module to determine how to best express it in NL.

When there is NL input from the dialogue partner, the dialogue manager sends that input and the concepts for the expected responses to the NL understanding module. We extended the dialogue manager so that if the NL input does not match any of the expected responses, then it next sends the NL understanding module the concepts for the initiations in the first step of every recipe. Thus the dialogue manager now sets up tiers of concept labels (i.e. language models) for concept classification where the first tier contains concepts that can potentially fulfill a discourse obligation and the second tier contains likely concepts for initiating a new topic.

When the human dialogue partner produces an initiation for a step, the dialogue agent must decide which of the expected responses to produce. It can randomly select one as with the current implementation or it can use the information the author supplies about correctness and deliberately pick a response that is wrong. In addition, since one of the possible responses is always an *unanticipated-response*, it can indicate that it doesn't know how to respond.

When the human dialogue partner produces an initiation that is the first step in a recipe, we call this topic initiative. We more broadly define topic initiative as any part of a

human dialogue agent's turn that matches a concept in the second tier of concepts as long as it was not previously recently matched according to the dialogue history. We make this exception because what was said earlier may have been repeated for increased coherency instead of being a show of initiative. Currently the extended system will always uptake a recognized initiative. If some of the concepts associated with the human partner's turn are in the first tier, and thus are expected, while some are from the second tier, the system will pursue responses to the first tier concepts and will later respond to those in the second tier. When the initiative topic is completed the system will attempt to resume whatever was interrupted. The dialogue manager makes use of discourse obligations to resume interrupted topics after topic initiative. If the human partner interrupts and then resumes an interrupted topic, the dialogue manager prunes any remaining obligations that arose during the interruption.

3. Future Work

There are two extensions that we intend to address next. First we need to add information about the relative importance of topics so that the agent can decide whether to accept, ignore or postpone topic initiative instead of always accepting it. If the dialogue agent's last turn is a step in a topic of equal or greater importance than the human's initiation then the human's initiative would be postponed. Otherwise it is accepted. If the topic is not identified then the initiative is ignored. To properly postpone, the system needs to first offer an acknowledgment of the initiative that communicates it has been understood and will be entertained once the current topic is completed as with "first answer my question and then we'll talk about <topic>". Then it should put the postponed initiative as a goal on the dialogue manager's agenda before retrying the current step. Finally, it must decide where to put the postponed initiative on the agenda.

The second addition addresses resuming an interrupted topic. We need to add information on the relationships between topics to aid in deciding whether to abandon or retry the unfulfilled step, as our model currently does, or to retry some higher level topic goal. Finally, we need to add an appropriate NL transition back to the interrupted topic to reduce the chance of an incoherency because of the topic shift. Once these two additional extensions are completed, we will evaluate our approximate model by comparing task performance and learning gains of students working with either it, a tutor-only version of the agent, or another student.

References

[1] R. Cohen, C. Allaby, C. Cumbaa, M. Fitzgerald, K. Ho, B. Hui, C. Latulipe, F. Lu, N. Moussa, D. Pooley, A. Qian, and S. Siddiqi. What is initiative? *User Modeling and User-Adaptive Interaction*, 8:171–214, 1998.

[2] B. Di Eugenio, P. W. Jordan, R. H. Thomason, and J. D. Moore. The agreement process: an empirical investigation of human-human computer-mediated collaborative dialogues. *International Journal of Human Computer Studies*, 53(6):1017–1076, 2000.

[3] M. Evens and J. Michael. *One-on-One Tutoring by Humans and Computers*. Lawrence Erlbaum Associates, Inc., 2006.

[4] P. W. Jordan, M. Ringenberg, and B. Hall. Rapidly developing dialogue systems that support learning studies. In *Proceedings of ITS06 Workshop on Teaching with Robots, Agents, and NLP*, 2006.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Mirroring of Group Activity to Support Learning as Participation

Judy KAY, Peter REIMANN, Kalina YACEF
School of Information Technologies
and
Research Centre on Computer-supported Learning and Cognition,
University of Sydney, NSW 2030, Australia

Abstract. Recent learning theories stress that learning does not necessarily translate solely into knowledge gains: rather, it can be measured in terms of increased participation and interaction of the individual with their group or community. We explore this in conjunction with exploitation of an existing professional tool, which we have enhanced with mechanisms to extract useful information from learner interaction traces and then make these available in a form that is meaningful for teachers and learners. This broad approach offers huge promise at a far lower cost than that needed to build pure intelligent learning systems. We present our learning environment used in a course where small teams of students work on software development projects and learn about working in groups. We report the success of the approach in terms of the different levels of interaction across two cohorts, the first using just the professional tool and the second using our enhanced visualisations in conjunction with it.

Introduction

Classic capstone computing courses are one important example of courses which require students to learn to work in small groups on a substantial project over one or more semesters. In this context, we want to support students in the very challenging tasks of learning to work effectively in a group, including, for example, how to share out work, communicate effectively, plan, monitor progress and learn to develop trust (Salas et al. 2005). The knowledge that students have about the project is a vehicle for learning these group processes, rather than the other way around.

A particularly appealing approach to supporting this learning about effective group work is to make use of existing sophisticated tools that are used in the workplace to support teams: the obvious appeal is that these are well suited to supporting aspects of working in a group and these tools already exist and are well maintained. However, to support effective learning about group work, we have attempted to exploit the huge amount of data that can be captured by such tools. To make this data useful, we need to be able to harvest it and then to make effective use of it. In the next section, we will present emerging understandings of the nature of learning indicating that knowledge gains can be observed in terms of increased participation and interaction of the individual with their group or community. We then describe how this has been operationalised in our Trac-Plus learning environment. We then report the results of its use over two cohorts of student groups and report our conclusions.

1. The TRAC-Plus Learning Environment

Our Trac-Plus learning environment is built around Trac/SVN and its data. We have complemented it with a set of mirroring visualisations depicting elements of the team members' activity and statistical and data mining functionalities.

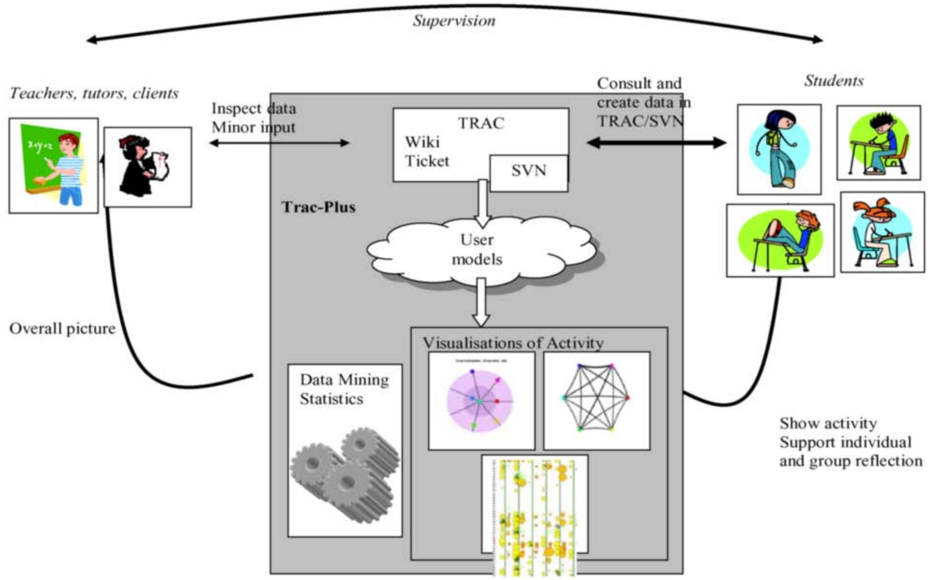


Figure 1. Trac-Plus learning environment

- *Trac software,* a state-of-the-art, free open source tool designed for programmers working in teams. Trac has a wiki for collaborative editing of web pages for general group communication, an issue tracking system based on so-called tickets and a browsing interface to a repository based upon the version control system called Subversion (SVN), for storing documents like source code, including any versions.
- *User model data* built using from the TRAC traces of students' actions, which we call *events.*
- *Visualisations of team activity,* aiming to mirror some aspects of the students' activity. They do not provide interpretative feedback and it is up to the users (students and teachers) to make sense of the information presented.
- *Data mining.* Further analysis can be carried out on the same data, in order to discover patterns that differentiate successful groups from less successful ones. If such information is detected early, feedback might help the group to improve.

2. Experiment and results

We use this environment in the 2005 and 2006 forms of a 13 week project course with teams of 5 to 7 students. The issues we explored were:

2.1. Do students participate more when they have access to the diagrams?
The difference when comparing the two cohorts is very striking. First, by looking at the Wattle Trees for each group in each cohort, the 2006 groups were undoubtedly more active than the 2005 across the three media. In fact, the top groups in each cohort were comparable (especially allowing for a heavier use of the wiki in 2006) but the average and weak groups showed dramatically more activity in the second year. For the critical Ticket medium, the 2006 class had triple the tickets in the previous year. The number of SVN files averaged 15% higher. The

wiki pages count also quadrupled in 2006 but we do not know how much of this increase is due to the use of wiki for a change requirement that the wiki be used for assessed reports.

2.2. Do students interact more when they have access to the diagrams?
Using a frequent sequential pattern algorithm (Kay et al. 2006) we looked at interaction patterns in 2005 and 2006. We found that students in 2006, who had regular access to the diagrams, interacted twice to 3 times as much as the ones in 2005.

2.3. Do students think the diagrams are useful for their group work?
We ran a survey of student attitudes to the three visualizations. All were rated useful, with a higher rating from team leaders and trackers. This is understandable as these roles involve work allocation and checking progress. On a scale of 1 (Disagree) to 7 (Agree), students were asked whether they "… recommend the use of these diagrams for future offerings or other group project offerings". The results showed a strong approving opinion, in particular from team managers and trackers.

2.4. How do students use the diagrams or why don't they?

To the question "I/my group changed something during the semester in light of what I/we saw in the diagrams", 40% of students answered "Yes". Those students explained that they used the diagrams for improving communication in the group, stepping up workload, or balancing workload among team members. For those who answered "no", the reasons relate to awareness of difficulties or lack of them in an effective, functioning group. These students' comments tell us that overall the visualisations, especially the Wattle Tree, helped them monitor and, if needed, regulate their participation. This is exactly what we intended them for.

2.5. Teachers and tutors impressions
For teachers, the visualisations played an invaluable role in helping groups. The teacher could come to a meeting with the group and use this as the starting point for the discussion of the progress of the group and its management. Serious problems, such as a drop in activity or surprising combinations of media use served as a basis for exploring the nature of the potential problem and beginning to address it early. The visualisations also served as a starting point for the teacher to check the actual media, such as the wiki pages that were being heavily changed.

3. Conclusion
Recent literature points to the importance of engagement and involvement as markers of learning. We have also described how we have taken this view as a basis for enhancing an existing group work tool, Trac, to mirror the activity within it. In terms of the observed increase in activity, the students views expressed in surveys, the teacher's views and experiences as well as the ways that students referred to the visualisations when discussing the operation of their groups, it seems that our Trac-Plus enhancement points to the promise of this approach to supporting groups to learn to be more effective.

References

Kay, J., N. Maisonneuve, K. Yacef and O. Zaiane (2006).Mining patterns of events in students' teamwork data. Proceedings of Educational Data Mining Workshop, held in conjunction with ITS'06. Taiwan.

Salas, E., D. E. Sims and C. S. Burke (2005). "Is there a "Big Five" in teamwork?" Small Group Research **36**(5): 555-599.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Open Learner Models: Opinions of School Education Professionals

Alice KERLY and Susan BULL
Electronic, Electrical and Computer Engineering,
University of Birmingham, Edgbaston, Birmingham, B15 2TT, UK
{alk584, s.bull}@bham.ac.uk

Abstract. This work surveys the opinions of teachers and other education professionals regarding the potential for Open Learner Models (OLM) in UK schools. We describe the aims of OLM, and of current UK initiatives involving formative assessment and promotion of metacognitive skills. We conclude that UK education professionals appreciate a synergy of these approaches, and that OLM-based systems could be valuable in achieving educational aims in schools.

1. Introduction

Open Learner Modelling (OLM) extends traditional learner modelling in Intelligent Tutoring Systems with the aim of making the model's contents a visible and interactive part of the learning environment. A key reason to open the learner model to students is to encourage them to reflect on their knowledge and learning process. Teachers may use OLMs to adapt teaching, for curriculum planning, or for formative assessment [1].

Reflective and metacognitive skills have long been recognised as important parts of the learning process [2]. Black and Wiliam [3] argued that formative assessment is an essential component of classroom work, but that current practices were weak. Modern UK educational practice now promotes the development of reflective skills to enhance learning. In the UK, the term 'Assessment *for* Learning' (AfL) is widely used to describe classroom practices aimed at promoting formative assessment. Pupil self-assessment is regarded as an essential component of this [3]. The aims of AfL (promoting reflection, using assessment to modify teaching, conducting pupil self-assessments and providing formative feedback) closely mirror the ethos of OLM.

Research on the opinions of teachers and other educational professionals towards the potential for the use of OLM in schools has so far been limited. Given the range of issues that OLM can support, the apparent close fit between OLM and AfL approaches, and the fact that these professionals will be crucial in OLM uptake, it seems timely that these school professionals be consulted. Therefore, we present a survey to elicit the opinions of teachers and educational professionals on the potential for OLM in schools.

2. Educational Professionals' Beliefs about the Potential for OLM in Schools

There were 15 participants, comprising UK school teachers (five Primary; three Secondary), two head teachers (Primary), and five Local Authority advisers or UK

Government (OfSTED) school inspectors. Participants were given documentation introducing OLM, including examples of six OLMs, discussing the aims of OLM, and summarising OLM research. Participants completed a questionnaire comprising questions with a 5-point Likert scale, and a number of free response questions.

	< strongly disagree-strongly agree >					Mean score
	(1)	(2)	(3)	(4)	(5)	
What would OLM be useful for?						
For teachers in assessing pupils	0	0	0	10	2	4.3
For feedback to pupils	0	0	2	8	5	4.2
For teachers' planning	0	0	5	7	3	3.9
Motivating pupils	0	0	2	6	7	4.3
What aspects would be useful to the learner?						
Viewing individual learner model	0	0	1	9	5	4.3
Comparing knowledge to expectation	0	0	0	11	4	4.3
Comparing their knowledge to peers	0	4	9	1	1	2.9
What aspects would be useful to the teacher?						
Viewing individuals' learner models	0	0	0	10	5	4.3
Comparing models with expectation	0	0	0	10	5	4.3
Comparing individuals to their peers	0	3	4	6	2	3.5
For what reasons would you want learners to access their model?						
Reflect & understand knowledge	0	0	1	10	4	4.2
To help them plan their learning	0	0	5	6	4	3.9
To give more control/responsibility	0	0	5	5	5	4.0
Practical Issues about OLM & AfL						
OLM fits with classroom teaching	0	1	8	3	2	3.4
OLM reflects educational philosophy	0	0	2	7	5	4.2
OLM supports AfL aims & objectives	0	0	3	9	3	4.0

The open ended portion of the questionnaire covered the following issues.
Asked about **particular subject areas that OLM would lend itself to**, some respondents felt all subjects could be appropriate (e.g. *"hard to think of a subject that couldn't benefit"*), some gave specific suggestions (e.g. *"D&T, Maths, Science"*, *"Subjects where answers are fact based"*), or broader views (e.g. *"OLM goes beyond subject areas...contribution in thinking skills, problem solving and metacognition"*).

Contributory **factors that would influence personal uptake/advocating of OLM** were related to the OLM tool (e.g. *"quality of the underlying tool", "quality of interpretation"*, and "*intuitiveness of the OLM software"*), and practical considerations (e.g. *"time involved for teachers"*, *"Whether OLM is tied with Key Stage material"*.

Respondents offered a range of uses for **OLM in assessment/feedback/planning**, including *"Feedback...for class, groups and 'outliers' could be used quite easily in planning for extension and remediation"*, *"I would use this as a whole class delivered task"*, *"To motivate pupils to plan their own development"*, and *"setting targets"*.

Respondents felt **OLM would be useful to children or teachers** in that "The greatest benefit to students would come from teachers [giving] the power over learning to their students - enter into a learning dialogue on the basis of the OLM feedback", "Build on prior achievement / understanding. Highlight gaps in learning". Two school inspectors stated "OLM…[makes] a strong contribution in such aspects as thinking skills, problem solving and metacognition" and "engaging pupils in their own learning …is a key towards faster progress and better understanding". Overall it was felt that

individual OLMs would be useful for children and teachers, with access to peer models less so. The main difficulties are in practical deployment rather than educational issues.

3. Analysis and Discussion

AfL and OLM concepts appeared to be accepted and valued by those surveyed (see open ended comments). Responses suggest that OLM could give students power over their learning, a learning dialogue and motivation – all supporting AfL as a method to actively engage students in their learning. The free responses suggest recognition of OLM's potential in developing non-subject specific skills to support learning.

Viewing their individual learner model or comparing their knowledge to teacher expectations was considered to be the most useful to children and teachers. Comparing knowledge to peers received a largely neutral response, though some respondents identified allowing students to compare their knowledge to peers as not being useful. Care needs to be taken to ensure motivation or self esteem are not damaged, that learning difficulties are treated sensitively, and that access is appropriate for the individual. The belief that viewing peer models may not be useful is interesting given results of prior studies (e.g. [4] where pupils aged 8-9 showed interest in comparing their progress with peers). The key reason for wanting learners to access their model was to help them better understand their knowledge and problems, helping them identify their needs themselves. Access to enable pupils to plan their learning or take more control and responsibility over learning was also considered desirable. Teachers would use OLM in particular for assessment, feedback and to motivate pupils.

Crucial to developing OLM use in schools, respondents agreed that OLM reflected their philosophy of learning, and the potential for OLM in achieving AfL aims. Guidelines for development include: integrating OLMs with schemes of work, targets or curricula to minimise teacher workload; OLMs must be quick to use and learn for teachers and students (but designs should not reduce complexity beyond usefulness).

4. Conclusion

This survey has identified that, while there are challenges to be met in enabling wider uptake of OLMs in schools, teachers and educational professionals see the potential benefits of OLM. The methods and aims of OLM are not only in keeping with, but actively support and promote current UK educational policy and practice.

References

1 Bull, S. & Kay, J. (in press). Student Models that Invite the Learner In: The SMILI Open Learner Modelling Framework. International Journal of Artificial intelligence in Education.

2 Dewey, J., How We Think: A Restatement of the Relation of Reflective Thinking to the Educative Process. 1933, Boston: DC. Heath and Company

3 Black, P. & Wiliam, D. Inside the black box: raising standards through classroom assessment, 1998. London: King's College London, School of Education.

4 Bull, S. & McKay, M. (2004). An Open Learner Model for Children and Teachers: Inspecting Knowledge Level of Individuals and Peers. Intelligent Tutoring Systems: 7th International Conference, Springer-Verlag, Berlin Heidelberg, 646-655.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Novel Tools for Assessing Student Discussions: Modeling threads and participant roles using speech act and course topic analysis

Jihie KIM, Erin SHAW, Grace CHERN, Roshan HERBERT
University of Southern California, Information Sciences Institute
4676 Admiralty Way, Marina del Rey, CA 90292 USA
{jihie,shaw}@isi.edu, {gchern,rherbert}@usc.edu

Abstract. This paper describes new techniques for assessing pedagogical discourse via threaded discussions that are based on an analysis of speech acts and course topics. The context is an undergraduate computer science course. We use speech act analysis to assess the effect of instructor participation on student participation and to create thread profiles that help identify threads characterized by agreement, debate, and unresolved issues, such as threads that may have unanswered questions. For the first time, we integrate an analysis of course topics to help identify discussions of particular interest or confusion.

Introduction

On-line collaborative discussions (OCDs) play an important role in learning. Making use of the resulting dialog to assess understanding is an open research problem. In previous work, we developed measures of discussion activities that relied on the quantity and quality of contributions and terms [3,5]. This paper extends our existing assessment measures [2a,2b,2c] to include an analysis of speech acts and course topics.

1. Classifying Student Contributions by Speech Acts

Our dataset consists of one semester of discussions from an undergraduate Operating Systems course at the University of Southern California. For conversation analysis, we defined a set of speech acts (SAs) [4] that relate pairs of messages in the dataset. We use the six SA categories shown in Table 1, derived by merging the original twelve [2a], using a Kappa value [1] to measure agreement between annotators. Table 2 shows the distribution of SAs in the set and examples of the cue words used to identify them.

Table 1: New speech act categories and descriptions

QUES	Question	ELAB	Elaboration
ANNO	Annotation	CORR/OBJ	Correction/Objection/Criticism
ANS/SUG	Answer/Suggestion	ACK/SUP	Support/Acknowledgement/Complement

Table 2: Distribution of speech act and surface cue words in dataset

Speech Act	Freq.	%	Sample Surface Cue Words
ACK/SUP	56	7.54	“good job” ”you got it” “agree” “good/nice/correct answer”
ANS/SUG	321	43.2	“perhaps” “how about” “you might” “you probably” “maybe”
CORR/OBJ	41	5.52	“doesn’t mean” “are you sure” “what/how about” “not work”
ELAB	53	7.13	“…and” “also” “by the way” “same question” “so…”
QUES	269	36.2	“how” “what” “can we” “are”/”is” “why” “was wondering”

2. Effect of Instructor Participation on Student Discussions

Table 3 shows the frequency of the instructor's SAs and Table 4 shows the effect of instructor participation on student participation. Although threads in which the instructor participated were longer, fewer students contributed to the threads. The results are consistent with our earlier findings. However, when we split these threads by SA types, specifically, those in which an answer was provided and those in which one wasn't, we found that providing scaffolding without answers had the effect of increasing both the number of students and the number of messages in a thread.

Table 3: Frequency of instructor speech acts

#Instructor SpeechAct	ACK/SUP	ANNO	ANS/SUG	CORR/OBJ	ELAB	QUES
	18	0	157	5	11	9

Table 4: Effects of instructor participation

		Average # Students	Average # Posts
Threads w/o instructor participation		2.91	3.09
Threads w/ instructor participation		2.60	3.36
Speech Act	ANS (plus other SAs)	2.55	3.29
	(No answers, other SAs only)	3.33	4.50

3. Creating Thread Profiles using Speech Act Analysis

We also analyzed threads with no instructor involvement, categorizing them by four pattern groups (A, B, C, D) as shown in Table 5. Each row represents a different thread pattern and a bracketed identifier represents a student; e.g., <P1> is the first contributor. Patterns in Group A depict simple information exchange. In Group B, interactions such as elaborations, corrections or further questions were needed to find the answer. Group C depicts agreement among participants and answer resolution. Group D cases have unresolved issues, e.g., in some threads, no answer was given and help was needed

Table 5: Thread profiles: patterns of student interactions without the instructor

Pattern Group A: short information exchange on non-controversial issues
16 QUES <P1> ANS <P2>
1 ANNO <P1> ACK <P2>
Pattern Group B: Discussion on somewhat complex issues, answers may have been found.
1 QUES <P1> ANS <P2> CORR <P3>
1 QUES <P1> ELAB <P2> ANS <P2>
1 QUES <P1> QUES <P2> ANS <P1>
1 ANS <P1> CORR <P2>
1 QUES <P1> ANS <P2> ANS <P1>
1 QUES <P1> ANS <P1> QUES <P2> ANS <P3>
1 QUES <P1> ELAB <P2> ELAB <P3> ACK <P4> QUES <P2> ANS <P4>
Pattern Group C: collaborative discussion on complex issues, followed by agreeable conclusion
1 QUES <P1> ANS <P2> QUES <P3> ANS <P2> CORR <P3> ACK <P2>
Pattern Group D: Students may have unresolved issues.
1 QUES <P1> CORR <P2>
1 QUES <P1> ACK <P2> *(repeated question answered in previous thread)
1 QUES <P1> ANS <P2> CORR <P1> ANS <P2> ANS <P3> QUES <P2>
3 QUES <P1> ELAB <P1>

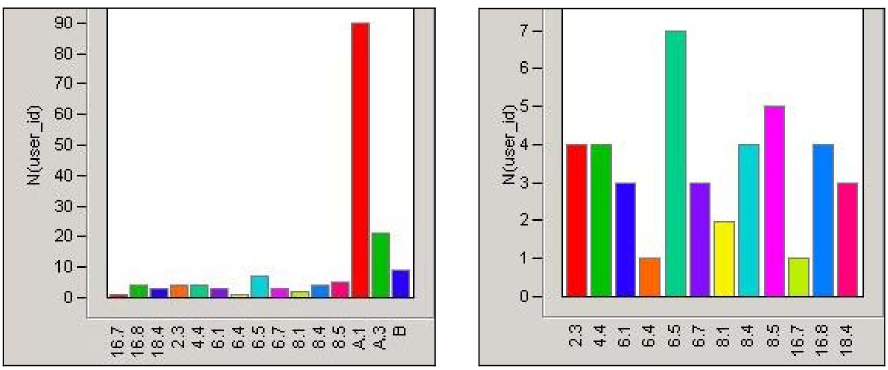
4. Integrating Course Topics to Identify Interest and/or Confusion

To assess topics of interest or confusion we used a set of course topics, shown in Table 6, from a course ontology induced from a textbook [2c]. The mapping from the data set to the ontology was done manually. Figure 1 shows two views of the topics discussed. On the left, students discuss administrative issues more than others; on the right, administrative threads are excluded. The topic SAs will tell us what students ask about.

Table 6: Topics used for analysis for this data set

Id	Section Description	Id	Section Description
4.4	Threads: Threading Issues	8.5	Main memory: Structure of page table
6.1	Process synchronization: Background	16.7	Distributed system structures: Robustness
6.4	Process synchronization: Hardware	16.8	Distributed system structures: Design issues
6.5	Process synchronization: Semaphores	18.4	Distributed coordination: Concurrency control
6.7	Process synchronization: Monitors	A.1	Project and exam information
8.1	Main memory: Background	A.3	General course information
8.4	Main memory: Paging	B	Posts with images and movies

Figure 1: Topics (by Id) discussed by number of users



5. Conclusion

We have extended our work on assessing pedagogical discourse by incorporating an analysis of speech acts and course topics. These models were used in the development a machine learning tool that automatically classifies speech acts [6].

Sponsor: National Science Foundation CCLI-Phase 2 program (Award No. 0618859).

References

[1] Cohen, J. (1960). A coefficient of agreement for nominal scales. Educational and Psychological Measurement, 20, 37-46.

[2a] Feng, D., Shaw, E., Kim, J., & Hovy E. (2006), An Intelligent Discussion-Bot for Answering Student Queries in Threaded Discussions. Proc. of Intelligent User Interfaces Conference.

[2b] Feng, D., Kim, J., Shaw, E., & Hovy E. (2006), Learning to Detect Conversation Focus of Threaded Discussions. Proc. of the Joint HLT Conf./Annual Mtg of the NAACL.

[2c] Feng, D., Kim, J., Shaw, E., and Hovy E. (2006), Towards Modeling Threaded Discussions through Ontology-based Analysis, Proc. of National Conference on Artificial Intelligence.

[3] Kim, J. & Beal, C (2006), Turning quantity into quality: Supporting automatic assessment of on-line discussion contributions, AERA.

[4] Searle, J. (1969). Speech Acts. Cambridge: Cambridge Univ. Press.

[5] Shaw, E. (2005). Assessing and Scaffolding Collaborative Learning in Online Discussion, Proc. of AI Ed.

Evaluating of a Scaffolding System for Supporting Less Experienced Feedback Givers

Duenpen KOCHAKORNJARUPONG and Paul BRNA
The SCRE Centre, University of Glasgow, St Andrews Building
11 Eldon Street, Glasgow, G3 6NH, Scotland, UK
duenpen@scre.ac.uk, paul.brna@scre.ac.uk

Abstract. This paper presents an evaluation of McFeSPA (Metacognitive[1] Feedback Scaffolding System for Pedagogical Apprenticeship), a system to help inexperienced teaching assistants (TAs)[2] who lack training in how to provide quality feedback to help them improve their feedback skills while marking programming assignments. The system exploits techniques drawn from artificial intelligence, cognitive psychology and theories of education. The results of this study indicate that McFeSPA could help TAs who mark programming assignments think about/reflect on their feedback giving to provide quality feedback to students in general.

Keywords. Learning to give feedback, scaffolding, feedback patterns, situated learning, metacognition

Introduction

In previous work [1], we introduced a framework based on providing a range of "feedback patterns", derived from [2], for teaching novice TAs to give quality feedback. These patterns can be combined with contingent learning [3] to help novice TAs become experienced teachers. In the Intelligent Tutoring System (ITS) and AIED communities [4] there has been no explicit use of feedback patterns – nor have they been employed in the context of training people to provide quality feedback. This research emphasizes the use of feedback patterns to construct feedback. Students, in particular, those in higher education, require quality feedback (e.g. more detailed feedback) to improve their learning. Training novice TAs to give quality feedback with computer-support is a challenging research issue requiring expert knowledge for giving such feedback and providing the levels of help to support giving quality feedback. Thus, we hypothesised 1) Providing content scaffolding (i.e. detailed feedback using contingent hints) in McFeSPA can help TAs increase knowledge/ understanding in the issue of learning to give feedback according to their perspective; 2) Providing

---

[1] In term of metacognitive scaffolding, we mean that McFeSPA provides more opportunity for fostering reflective thinking by the feedback giver about giving feedback.
[2] Inexperienced TAs are taken to include novice teachers, novice tutors, and novice lecturers

metacognitive scaffolding[3] in McFeSPA can help TAs reflect/rethink their skills in giving feedback; 3) When the TAs obtain knowledge about giving quality feedback, the approach taken by McFeSPA of providing adaptable fading allows the TAs to learn alone without any support. In this paper, we present the usability and evaluation of the system, discussion, and draw out our future work.

1. Evaluation of McFeSPA, and Results

Here, we summarise the usability evaluation then describe the main evaluation in more detail. For the usability evaluation, we used multiple units of analysis from multiple sources of evidence including log data, semi-structured interviews, satisfaction questionnaires, and a think aloud protocol. The three evaluators, including an expert evaluator, were able to understand and operate the mechanisms incorporated into the prototype. Most recommendations were aimed at improving the interface. Some, however, did impact on the pedagogical approach e.g. an encouraging message pops up when the users give the right feedback for each action – but all the evaluators suggested removing these messages because they felt annoyed with them. We now outline the main study.

- *Data collection:* we used multiple methods to reduce inappropriate certainty [5] and to clarify ambiguous data [6]: a combination of a form of cognitive walkthrough [7], thinking aloud [8], structured interview, questionnaire, and a pre-test/post-test comparison. Two TA groups tested the system: an experimental group who first used the system with offers of help and then without any offer of help, and the comparison group which received no offers of help.

- *Results:* The results were examined based on a triangulation approach [9] containing many sources for data collection – i.e. observation checklist, log-files, questionnaire, interview, pre-test, post-test. Overall activities took about two hours on average for each participant. All six participants agreed that the basic idea of McFeSPA was helpful and they enjoyed learning with McFeSPA. All participants agreed that they can apply the knowledge they obtained by using the system to mark any programming assignment (not only Prolog). All of them agreed that McFeSPA helped them reflect/rethink their skills in giving feedback which include 1) explaining what was wrong, 2) the strengths of automatic analysis of students' error, 3) giving more structure (help to see the mistake), 4) recognizing more than one type of feedback, 5) measurement of McFeSPA's skill meter was helpful, 6) deciding which feedback to give/which type of error the student made, and 7) encouragement to give more detailed feedback. Overall, we found that our participants learned differently by using McFeSPA. According to our hypotheses, the evaluation study indicates that there were some TAs who improved their feedback giving. Not all, because they are reasonably experienced TAs and some had previously been trained in giving feedback. One of the experimental group had a need to access a help-file to help them use the system. This suggests that even for adults there is a need to encourage help seeking [10].

Analysis of the results of this study indicates that McFeSPA helps TAs who mark programming assignments think about/reflect on their giving feedback to provide quality feedback to students in general. There is some promise for the approach of McFeSPA and some for the implementation in the following discussion.

---

[3] Metacognitive scaffolding refers to the different messages about feedback which feature in the contingent hints provided. There are also pop up messages that reference how the feedback report is progressing.

2. Discussion and Future work

In the current state, we evaluated the usability of the prototype and the potential the approach for helping real TAs to improve their feedback giving. The system was implemented in a manner sufficient to test the hypotheses. Most results we obtained correspond with our model even though some aspects of the implementation are not consistent with the users' wishes. In general, the system helped to show that giving feedback to the feedback giver has benefit. However, experienced TAs wanted the system to help them to mark quickly with convenient tools supported properly in real use. Their requirements correspond with the design of the system but some do not correspond with the implementation. We need to take into account their comments so that the next version McFeSPA could support all the needs of both novice and experienced TAs.

Improving the various kinds of support will be part of our future work. We believe that this research makes a novel contribution to the field of Artificial Intelligence in Education by focusing on how to train people to give quality feedback while marking assignments, in our case, in the context of teaching programming. McFeSPA helps TAs directly, and students indirectly. After further improvements, our long term aims include the development of McFeSPA to provide scaffolding for a range of ILEs and also to provide services for web-based systems.

References

[1] Kochakornjarupong, D., Brna, P., and Vickers, P. (2005). Who Helps the Helper? A situated Scaffolding System from Supporting Less Experienced Feedback Givers. In Proceedings of 12th International Conference on Artificial Intelligence in Education, Amsterdam, The Netherlands, 18-22 July 2005.

[2] Eckstein, J., Bergin, J., and Sharp, H. (2002). Patterns for Feedback. In Proceedings of EuroPLoP 2002. Seventh European Conference on Pattern Languages of Programs, Irsee, Germany.

[3] Wood, D. (2001). Scaffolding, contingent tutoring and computer-supported learning. International Journal of Artificial Intelligence in Education, 12, 180-292.

[4] Kumar, V., McCalla, G., and Greer, J. (1999). Helping the Peer Helper. In Proceedings of International Conference on Artificial Intelligence in Education, Le Mans, France.

[5] Robson, C. (2002). Real world research. Oxford: Blackwell.

[6] Molka-Danielsen, J. (2000). System Success and Failure: Implementation of IT. Available: http://home.himolde.no/~molka/in320/t00c13.html [Accessed on 8 June 2006].

[7] Rieman, J., Franzke, M., and Redmiles, D. (1995). Usability Evaluation with the Cognitive Walkthrough. In Proceedings of CHI'95 MOSAIC OF CREATIVITY.

[8] Ericsson, K.A. and Simon, H.A. (1993). Protocol analysis: verbal reports as data (Rev. ed.). Cambridge, Mass: MIT Press.

[9] Yin, R.K. (1994). Case Study Research: Design and Methods (2nd ed. Vol.5). London: sage.

[10] Aleven, V., et al. (2003). Help Seeking and Help Design. Review of Educational Research, 73(3), 277-320.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

The Effect of Open Student Model on Learning: A Study

Amruth KUMAR, Adrian MARIES
Ramapo College of New Jersey,
505, Ramapo Valley Road, Mahwah, NJ 07430, USA
{amruth, amaries}@ramapo.edu

Abstract. We conducted a within-subjects study of the effect on student learning of using open student model in the form of taxonomic concept map. We found a significant improvement in learning on two questions related to the topic of the tutor, and no pre-post improvement on questions unrelated to the topic of the tutor. These preliminary results suggest that students may learn concepts indirectly by viewing the open student model presented as taxonomic concept map after each problem.

Keywords. Concept maps, Open student model, Intelligent tutors, Programming

Introduction

Making the student model available to students can make them aware of their own knowledge, or lack of it, and, in turn, improve their learning [3,4]. One survey shows that students want to have access to their learner models [1]. If the student model is made available in a table format though, it is often difficult to understand [6]. Therefore, visualization of data is a critical part of the open student model [7]. Concept maps, with their ability to reveal the relationships among concepts, have been proposed as a mechanism to present the learner model [1,2,5,8].

Can students learn concepts by simply viewing the open student model presented as a concept map after each problem, especially concepts that are not explicitly mentioned in the problems on which they are tutored? In fall 2006, we conducted a within-subjects evaluation to answer this question. We used a tutor on arithmetic expressions for our evaluation. It presents problems on evaluating arithmetic expressions (e.g., 5 + 4 % 8), and provides demand feedback.

1. Evaluation

We used the traditional pre-test-practice-post-test protocol to evaluate the hypothesis. First, the subjects answered a pre-test consisting of 9 questions, 6 related and 3 unrelated to arithmetic expressions. Next, they worked with the tutor for 15 minutes solving expression evaluation problems and reading the feedback. After solving each problem, the subjects were shown their open student model as a taxonomic concept map (domain concepts are nodes, links are is-a and part-of relations, and the map is an and-or tree), with their percentage completion of each concept graphically displayed in

the corresponding node. Finally, the subjects answered a post-test consisting of the same 9 questions as on the pre-test.

The subjects were students in a Psychology course who used the tutor over the web, on their own time, as part of the requirements for a Psychology course. The questions on the pre-test and post-test were:

1. How many arithmetic operators are available in C++ programming language? 1/2/3/4/5/6
2. Pick ALL the operators among the following that are C++ arithmetic operators (check all that apply): <, * , !, /, %, ^
3. How many of the following numbers are prime: 2, 3, 4, 5, 6, 7, 8, 9, 10.
4. Which of the following C++ operators have 'integer' and 'real' types (check all that apply)? - <=, >, &&, +, /, ^
5. For which of the following C++ operators is 'Dividing by Zero' an issue (check all that apply)? - >=, ||, %, ^, !
6. What is the sum of the internal angles of a triangle? 90/180/270/360/450/600
7. To how many C++ arithmetic operators does the issue of 'Precedence' apply? – none/only one/all/only two/only three
8. Pick all the issues that apply to all the C++ arithmetic operators (check all that apply) – Coercion, Correct evaluation, Error, Associativity, Real
9. Which of the following are types of operators in C++ (check all that apply)? – Relational, Abstraction, Repetition, Logical, Selection, Assignment

Questions 3, 6 and 9 are not related to arithmetic expressions, and were meant to serve as control questions for each subject. Answers to questions 1,2,4,5,7,and 8 cannot be synthesized without an overview of the domain of arithmetic operators, such as that provided by a concept map, since these questions are not about any particular operator, but rather about groups of operators. During the problem-solving session, the tutor never explicitly provided the answers to any of the questions. But, the answers to questions 1,2,4,5,7,and 8 were evident from examination of the open student model presented as a taxonomic concept map after each problem.

2. Analysis

23 students participated in the evaluation. We used negative grading to penalize guessing – if a problem had n answering options, m of which were correct, the student received $1/m$ points for each correct answer and lost $1/(n - m)$ points for each incorrect answer. For each subject, we combined the scores on all the related questions (Questions 1,2,4,5,7, and 8) and the scores on all the unrelated questions (Questions 3,6, and 9). We did a 2 X 2 within-subjects ANOVA analysis of the collected data with pre-post and related-unrelated as the two within-subjects factors. We found a significant main effect for related versus unrelated questions [$F(1,22) = 6.725$, $p = 0.017$] – students scored 1.833 points on related questions and 1.238 points on unrelated questions. We found a significant main effect for time-repeated measure (pre-post) [$F(1,22) = 7.659$, $p = 0.011$] – students scored 1.246 points on the pre-test and 1.825 points on the post-test. We found a significant interaction between related versus unrelated and time repeated measure [$F(1,22) = 16.754$, $p = 0.000$] – whereas the pre-post change on related questions was from 1.175 points to 2.491 points, the same on unrelated questions was from 1.318 points to 1.159 points.

We did a post-hoc analysis by question. We found the pre-post changes on only questions 2, 5 and 6 to be statistically significant at the $p < 0.05$ level. On question 6, which is unrelated to the topic of the tutor, the average *decreased* from 0.782 to 0.652. Among the related questions, on question 2, the average improved from 0.202 to 0.694. The correct answer to this question was *, /, and %. On Question 5, the average improved from 0.043 to 0.565. The correct answer to this question was %, which is an operator unique to programming languages.

In both the questions, it is reasonable to argue that students selected only the operators on which they had solved problems. But, not all the students solved problems on all the arithmetic operators during practice. Most of the students did not solve any problem on dividing by zero error with % operator. Moreover, students had no basis to rule out the distracters in either question, except by inspecting the concept map. By using negative grading, we discounted for random guessing. Since the subjects were unaware that we were using negative grading, this could not have prompted them to be conservative in their answers and select only the operators on which they had solved problems. Finally, since the questions were of an overview nature, deducing answers to them from individual problems required more synthesis and recall than could be expected of Psychology subjects using the tutor for a not-for-grade exercise.

Our preliminary conclusion is that students may learn some concepts that are not explicitly covered by a tutor, simply by examining the open student model presented as a taxonomic concept map of the domain. A confound of our evaluation is that subjects could have scored partial credit on the test questions by selecting only the operators on which they had solved problems. Therefore, further evaluations are needed to confirm/refute this conclusion. If confirmed, this would be another reason in favor of presenting the open student model as a taxonomic concept map of the domain.

Acknowledgements: Partial support for this work was provided by the National Science Foundation under grants CNS-0426021 and DUE-0631738**.** This work was supported in part by a grant from the Ramapo Foundation.

References

[1] Bull S., 2004. Supporting learning with open learner models. *Proceedings of fourth Hellenic Conference on Information and Communication Technologies in Education*, Athens, Greece, pp. 47-61.

[2] Bull S., Mangat M., Mabbott A., Abu Issa A.S., Marsh J., 2005. Reactions to inspectable learner models: seven year olds to University students. *Proceedings of Workshop on Learner Modeling for Reflection, International Conference on Artificial Intelligence in Education*, pp. 1-10.

[3] Dimitrova, V., Brna, P., and Self, J. A. 2002. The Design and Implementation of a Graphical Communication Medium for Interactive Open Learner Modelling. In *Proceedings of the 6th International Conference on Intelligent Tutoring Systems* (June 02 - 07, 2002). S. A. Cerri, G. Gouardères, and F. Paraguaçu, Eds. LNCS vol. 2363. Springer-Verlag, London, 432-441.

[4] Kay, J. (1997). Learner Know Thyself: Student Models to Give Learner Control and Responsibility, in Z. Halim, T. Ottomann & Z. Razak (eds), *Proceedings of International Conference on Computers in Education*, Association for the Advancement of Computing in Education (AACE), 17-24.

[5] Mabbott A., Bull S., 2004. Alternative views on knowledge: presentation of open learner models. *Seventh International Conference of Intelligent Tutoring Systems*. Springer, Berlin, Heidelberg, pp. 689-698.

[6] Mazza, R. and Dimitrova, V. 2004. Visualizing student tracking data to support instructors in web-based distance education. In *Proceedings of the 13th Iinternational World Wide Web Conference,* (New York, NY, USA, May 19 - 21, 2004). WWW Alt. '04. ACM Press, New York, NY, 154-161.

[7] Mazza R. and Milani C. Exploring Usage Analysis in Learning Systems: Gaining Insights From Visualisations. In: *Workshop on Usage analysis in learning systems. 12th International Conference on Artificial Intelligence in Education* (AIED 2005). Amsterdam, The Netherlands. 18 July 2005. 65--72.

[8] Zapata-Rivera and Greer, 2004. Interacting with inspectable bayesian student models. *International Journal of AI in Education*. vo1 4 (2). 127-163.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

The Effects of Error-Flagging in a Tutor on Expression Evaluation

Amruth KUMAR[a] and Peter RUTIGLIANO[b]
[a]*Ramapo College of New Jersey, Mahwah, NJ 07430, USA*
[b]*New York University, New York, NY*

Abstract. We evaluated the effect of providing error-flagging as support for error detection, but not error correction while the student is solving a problem. We found that providing error-flagging in addition to demand feedback during practice learning was no more effective than providing only demand feedback when the tutor did not explicitly mention that errors were being flagged. On the other hand, explaining and providing error-flagging without demand feedback during pre- and post-tests resulted in significantly better scores on pre- and post-tests even though error-flagging did not provide any error-correction support.

Keywords: Demand feedback, Error-flagging, Expression evaluation, Error detection, Error correction

Introduction

Demand feedback, error-flagging and immediate feedback are three typical types of feedback provided by tutors. When the different types of feedback provide error detection and error correction, they treat them differently: immediate feedback detects and suggests corrections for errors, whereas demand feedback expects the learner to both detect and correct errors. Error-flagging is a via-media – it detects errors for the learner, but leaves it to the learner whether and how to correct the error.

How does error-flagging stack up against demand feedback? One of the studies with the ACT Programming Tutor [1] found that immediate feedback helped learn the fastest, followed by error-flagging and demand feedback. There was little difference among these groups on tests. Another study with the SQL-Tutor [2] found one of the highest initial learning rates with error-flagging – measured in terms of the probability of violating a constraint after feedback had been provided on the constraint on prior occasions.

We studied error-flagging against demand feedback. Our study differed from previous studies in the following ways:

- We treated error-flagging as a support mechanism for error-detection only, e.g., in the ACT programming tutor, a student can request advice on an error-flagged step. In our study, error-flagging did not provide any error-correcting advice.
- We provided error-flagging feedback while the student was attempting the problem and demand feedback after the student had submitted the answer – in other words, we compared demand feedback versus error-flagging provided *in addition to* demand feedback.
- Finally, we considered the effect of error-flagging during testing.

In the next section, we will briefly describe the tutor on arithmetic operators and the two types of feedback provided by it: demand feedback and error-flagging. In section 2, we will describe the evaluation protocol and results of analysis.

1. Tutor on Arithmetic Operators

The tutor on arithmetic operators covers addition, subtraction, multiplication, division and remainder operators in C/C++/Java/C#. For these operators, it covers the concepts of precedence, associativity, correct evaluation of operators (especially operators such as remainder and integer division), and potential errors such as divide-by-zero, inapplicability of modulus operators to real operands in C++, incompatibility of boolean and numeric operands in Java/C#, etc.

The tutor presents arithmetic expressions and asks the student to evaluate them step-by-step, i.e., one operator at a time. In order to evaluate an operator, the student must click and drag the mouse across the sub-expression that involves that operator. In response, the tutor draws an under-brace to span the selected sub-expression, and pops up a dialog box to enter the intermediate result. After the student enters the intermediate result in the dialog box, the intermediate result is displayed immediately below the under-brace, at its center. The student must repeat this process for each operator/sub-expression/step in the expression. The student always has the option of undoing the last step, i.e., undoing the last under-brace and intermediate result, or clearing all the previously entered steps.

Once the student has entered his/her answer, whether partial or complete, the student submits the answer by clicking on the Submit button. Thereupon, the tutor provides demand feedback – it lists how many steps the student solved correctly. It displays the correct evaluation of the expression using under-braces and intermediate results. In addition, it prints a text explanation for each step, such as "16 / 5 returns 3. Since both the operands are integers, integer division is performed. Any fraction in the result is discarded." Since the student's attempt is displayed in the left panel and the correct evaluation is displayed in the right panel simultaneously, the student can compare the two solutions.

When the tutor is set to provide error-flagging feedback, as soon as the student draws an under-brace or enters the intermediate result, the tutor displays them in red if incorrect and green if correct. The student has the option to correct the error, since the undo option is always available to the student. But, the tutor does not provide any feedback as to why a step is incorrect. So, the tutor provides error-detection but not error-correction support. In a multiple-choice question, such support for error-detection but not correction could encourage the student to employ a trial-and-error approach to answering questions. But, in expression evaluation, if an expression includes n operators, there are n! options for the sequence of under-braces and infinite options for intermediate results. So, trial-and-error strategy does not pay off even when error-detection (but not error-correction) support is provided. If the tutor is not set to provide error-flagging, it displays all the under-braces and intermediate results in black.

2. Evaluation

In spring and fall 2006, we conducted web-based evaluations of the tutor. We used a between-subjects design, randomly dividing into two groups, the classes that used the tutors. We used the pre-test-practice-post-test protocol for evaluation:

- **Pre-test** consisted of 22 problems, to be solved in 7 minutes. The tutor administered the pre-test, and did not provide any feedback after each answer was submitted. The tutor used the pre-test to initialize the student model.
- **Practice** was adaptive – the tutor presented problems on only those concepts on which the student did not correctly solve problems during pre-test. After the student submitted the answer to each problem, the tutor provided demand feedback including whether the student's answer was correct, and the correct solution to the problem. Practice session lasted 15 minutes or until the student learned to correctly solve problems on all the concepts, whichever came first.
- **Post-test** consisted of 22 problems similar to those on the pre-test, and presented in the same order. The post-test was limited to 7 minutes. The tutor administered the post-test online, and did not provide any feedback after each answer was submitted.

The three stages: pre-test, practice and post-test were administered back-to-back, with no break in between. The students did not have access to the tutor before the experiment.

In spring 2006, we wanted to evaluate the effect of providing error-flagging in addition to demand feedback during practice. So, the control group received demand feedback, and the test group received error-flagging plus demand feedback during practice. However, when providing error-flagging, the tutor did not explicitly mention that if a step is marked in red, it is incorrect, and if a step is marked in green, it is correct. Neither group received any feedback during the pre-test and post-test. These groups have been designated groups A and B in Table 1.

Table 1: Summary of the treatments

Semester	Group Name	Type	Pre-Test	Practice	Post-Test
Spring 2006	A	Control	None	Demand	None
	B	Experiment	None	Error-Flag + Demand	None
Fall 2006	C	Control	None	Demand	None
	D	Experiment	Error-Flag	Error-Flag + Demand	Error-Flag

In fall 2006, we wanted to evaluate the effect of providing error-flagging not only during practice, but also during the pre-test and post-test. So, the control group received demand feedback during practice, and no feedback during pre-test and post-test. The test group received error-flagging plus demand feedback during practice. During pre-test and post-test, it received error-flagging feedback while the student was attempting to solve the expression, but no demand feedback after the student had submitted the answer. The tutor explicitly mentioned that errors were being flagged with color. These groups have been designated groups C and D in Table 1.

We conducted a 2 X 2 mixed factor ANOVA analysis on the problems solved, the total score and the score per problem, with pretest-post-test as the repeated measure, and the type of treatment as between-subjects factor. Our findings for spring 2006 were for groups A and B:

- There was a significant main effect for pre-test versus post-test on the number of problems solved [$F(1,111) = 237.663$, $p = 0..000$], the total score [$F(1,111) = 171.162$, $p = 0.000$] and the score per problem [$F(1,111) = 40.596$, $p = 0.000$].
- There was no significant main effect for the type of treatment and no significant interaction betweeen pre-post and the type of treatment.

Our findings for fall 2006 were for groups C and D:

- There was a significant main effect for pre-test versus post-test on the number of problems solved [$F(1,107) = 102.449$, $p = 0..000$], the total score [$F(1,107) = 93.74$, $p = 0.000$] and the score per problem [$F(1,107) = 35.603$, $p = 0.000$].
- There was a significant main effect for the type of treatment on the total score [$F(1,107) = 13.228$, $p = 0.000$], and the score per problem [$F(1,107) = 28.918$, $p = 0.000$], but not the number of problems solved [$F(1,107) = 2.037$, $p = 0.156$].
- There was no significant interaction betweeen pre-post and the type of treatment.

The average and standard deviation of the total score and score per problem for groups A-D are summarized in Table 2. We found no statistically significant difference in the number of problems solved by the four groups during either the pre-test or practice.

Table 2: Total score and score per problem

Semester	Type of Treatment	Pre-Test Score		Post-Test Score	
		Total	Per problem	Total	Per problem
Spring 2006	Group A	5.652	0.549	10.554	0.692
	Group B	4.996	0.542	10.470	0.661
Fall 2006	Group C	5.441	0.554	10.019	0.706
	Group D	8.074	0.772	13.655	0.878

Spring 2006 results indicate that providing error-flagging in addition to demand feedback during practice learning is no more effective than providing only demand feedback if the tutor does not explicitly state that errors are being flagged. This could be because: 1) Absent an explicit explanation of error-flagging, the students may not have realized the purpose of color-coded steps on their own. 2) The students were not required to fix errors before proceeding to the next problem. So, students may have simply ignored the feedback.

Fall 2006 results indicate that providing error-flagging during pre- and post-tests results in significantly better scores on pre- and post-tests even though error-flagging did not provide any error-correction support (effect size = 0.65 for the total score and 0.64 for the score per problem.) This was because students with error-flagging support chose to revise their answer far more often (325 revisions on pre-test, 422 on post-test) than those without error-flagging support (70 revisions on pre-test, 71 on post-test). The students with error-flagging did not randomly guess the correct revisions because: 1) they would have spent more time per problem doing so, and would have solved fewer problems, which is not true; 2) the increased score from the pre-test would not have carried over to the post-test, which it has. So, does the increased score signal increased learning? We plan to conduct delayed post-test (without error-flagging) to test this hypothesis.

Acknowledgements: Partial support for this work was provided by the National Science Foundation under grants CNS-0426021 and DUE-0631738**.** This work was supported in part by a grant from the Ramapo Foundation.

References

[1] Corbett, A.T. and Anderson, J.R. Locus of feedback control in computer-based tutoring: impact on learning rate, achievement and attitudes. Proceedings of the SIGCHI conference on Human factors in computing systems, 245 - 252 , 2001
[2] Mitrovic, A., Martin, B. Evaluating Effectiveness of Feedback in SQL-Tutor. IWALT 2000, December 4-6, 2000, Palmerston North, pp. 143-144.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Paragraph Development Schemata to Support English Composition

Hidenobu KUNICHIKA[a, 1] and Akira TAKEUCHI[b]
[a] *Dept. of Creation Informatics, Kyushu Institute of Technology, JAPAN*
[b] *Dept. of Artificial Intelligence, Kyushu Institute of Technology, JAPAN*

Abstract. We propose "paragraph development schemata" in English to help learners learn good paragraph structures for English composition. Each schema is expressed by standardized and fine grain sized terminology. Owing to the schemata, learners will be able to know what paragraph structures exist, recognize how each structure is different, and when a learner selects one schema she can be aware of the lack of necessary information by herself.

Introduction

The purpose of this study is to make learners good writers in English. Then, one question appears: what is "a good writer"? At least, they should be able to both (1) write grammatically correct English sentences, and (2) construct understandable logical structures. So far, researchers, including us, have paid great attentions to achieve (1) [1, 2] in comparison with (2). In this paper, we describe how important to achieve (2) is, and knowledge for both learners and computers to support (2).

When we use different language from the language we usually use, we will change not only "word" and "grammar", but also "concept", "thinking style", "structure of sentences", and so on. There are many good understandable logical structures depending on languages. So, to be a good writer in English, we should understand what logical structure is understandable in English, what thinking style (presentation style) is acceptable, and so on. Before that, of course, we should have clear awareness that they may not be the same as them in other language.

We have been focusing structures of "paragraph" which is a minimum unit of group of sentences in English. The paragraph is clearly defined, the structure of a paragraph is discussed, and paragraph writing is trained in English speaking countries. When we look for instructions and information on paragraph writing, we can get many information and text books (e.g., [3]). It is good, but brings new problems at the same time: how many types of structures are there? What grain size is easy to understand? How does structure-X differ from structure-Y? It is difficult to understand explanations of the structure, because there are only a few examples of the structure. These definitions still have ambiguity because they are written by natural language. To help learners learn paragraph writing, we extract some "paragraph development schemata (in short, PDS)" from many instructional information and text books for paragraph writing. Since to express paragraph structures as schemata, we can reduce their ambiguity, and translate them from schemata to computer-readable forms.

[1] 680-4 Kawazu, Iizuka, Fukuoka, 820-8502 JAPAN; E-mail: kunitika@ai.kyutech.ac.jp

1. Paragraph Development Schemata

PDSs express typical structures and patterns of English paragraphs as schemata. There are many PDSs depending on purposes we want to write in the paragraph. Each PDS consists of what should be written and the order. We have also picked up information for writing a good paragraph and classified it into *explanations* of each component in the schema, *tips* on writing, and *words and phrases*.

We have extracted 10 types of purposes of paragraphs and expressed each structure corresponding to each purpose as a schema. Table 1 shows the outline of each schema. In the expressions of schemata, 0, 1, * and + express the number of repetition. * and + mean "zero or more" and "one or more", respectively.

Table 1. The Types of Paragraph Development Schema

purpose	*definition*	*Schema*
Listing	Listing and explaining important items or categories on a topic	(Introductory sentence)^{0,1}, (Topic sentence), (Word for enumeration, Item, (Explanation of the item)*)$^{+}$, (Concluding sentence)
Illustration	Illustrating a topic	(Introductory sentence)^{0,1}, (Topic sentence), (Word for enumeration, Example, (Explanation of the example)*)$^{+}$, (Concluding sentence)
Comparison/ Contrast	Explaining similar/difference items between two topics or advantages and disadvantages of one topic	(a: *Block organization*) (Introductory sentence)^{0,1}, (Topic sentence), (Item on topic A, (Explanation of the item)*)$^{+}$, (Word for Transition of viewpoint), (Item on topic B, (Explanation of the item)*)$^{+}$, (Concluding sentence) (b: *Point-by-Point organization*) (Introductory sentence)^{0,1}, (Topic sentence), (Word for enumeration, Item on topic A, (Explanation of the item)*, Item on topic B, (Explanation of the item)*)$^{+}$, (Concluding sentence)
Analysis	Explaining components and their features of a topic	(Introductory sentence)^{0,1}, (Topic sentence), (Word for enumeration, Component, (Explanation of the component)*)$^{+}$, (Concluding sentence)
Analogy	Explaining items on a topic by using analogy	(Introductory sentence)^{0,1}, (Topic sentence(=describing a comparison)), (Word for enumeration, Item, (Explanation of the item)*)$^{+}$, (Concluding sentence)
Cause and Effect	Listing and explaining effects of a cause	(Introductory sentence)^{0,1}, (Topic sentence(=describing a cause)), (Word for enumeration, Effect, (Explanation of the effect)*)$^{+}$, (Concluding sentence)
Effect and Cause	Listing and explaining causes of an effect	(Introductory sentence)^{0,1}, (Topic sentence(=describing an effect)), (Transition sentence), (Word for enumeration, Cause, (Explanation of the cause)*)$^{+}$, (Concluding sentence)
Definition	Defining a topic by explaining items on the definition	(Introductory sentence)^{0,1}, (Topic sentence(=Defining a topic)), (Word for enumeration, Item, (Explanation of the item)*)$^{+}$, (Concluding sentence)
Classification	Classifying a topic into several categories from a viewpoint, and explaining each category	(Introductory sentence)^{0,1}, (Topic sentence), (Word for enumeration, Category, (Explanation of the category)*)$^{+}$, (Concluding sentence)
Opinion and Reason	Describing opinion on a topic and the reasons	(Introductory sentence)^{0,1}, (Topic sentence(=Describing opinion)), (Word for enumeration, Reason, (Explanation of the reason)*)$^{+}$, (Concluding sentence)

2. Details of Paragraph Development Schemata: An Example

As an example, we present details of the PDS for Comparison / Contrast paragraph. As shown in Table 1, there are two types of *Schema*: one is (a) Block organization and the other is (b) Point-by-Point organization. Selecting one of schemata depends on the number of items or the contents of a paragraph. For example, when many items are described, Point-by-Point organization is suitable. *Dependence* expresses such dependence relationships as follows.

- The number of items >= 5 -> Organization=Point-by-Point
- Aim=advantages & disadvantages of one topic -> Organization=Block

Explanation expresses what should be described in each component of the schema. *Explanation* contains general explanations and dependent explanations on this type. *Tip* describes important points for writing a good paragraph and/or matters that require attentions. One of tips on this type of paragraph is as follows. The component of the schema which relates to each tip is placed between brackets.

- You should write the same supporting items of topic B in the same order as those for topic A. (Item on topic B of Block organization)

Word keeps information on words frequently used in this type of paragraph. The information consists of words and phrases, the part of speech, the element in which the word is used in, restrictions in use, and dependence relationships. In the following examples, "{}" is used for expressing a set of words and phrases, "()" for parts of speech, "[]" for restrictions, and "->" for dependence relationships.

- Words used in this paragraph type (excerpts)
  - Aim=comparison -> {comparison(n), similarity(n), …}[]
  - Aim=comparison & Organization=Block -> {in comparison(p, n), likewise(adv), similarly(adv), …}[]
- Words used in general (excerpts)
  - true -> {Enumeration=first -> {first(adv), second(adv), third(adv), …}[Order*1],
    Enumeration=firstly -> {firstly(adv), secondly(adv), thirdly(adv), …}[Order],
    …}[Selection=Exclusive*2, Component=Supporting sentences]
  - Enumeration=next -> {next(adj), subsequent(adj), following(adj), …}
    [Selection=Exclusive*3, Component=Supporting sentences]

*1 Using each word in this order.
*2 Not switching from one set to another in the middle of a list.
*3 Trying to avoid using a word in multiple time.

3. Conclusions

We described the importance of learning paragraph structures and PDSs used for helping it. We introduced ten types of PDSs, but they are not exhaustive. As future tasks, we elaborate PDSs and construct an English composition support system with the PDSs. Moreover, we will consider how to support learning structures not only within a paragraph, but also inter paragraphs.

References

[1] Gamper, J., Knapp, J. (2001) A review of intelligent CALL systems, Comput. Assist. Lang. Learn., Vol.15, No.4, pp.329-342.
[2] Kunichika, H., Hirashima, T., & Takeuchi, A. (2006) Visualizing Errors for Self-correcting Discrepancy between Thinking and Writing, Proceedings of ICCE06, pp.535-542, Beijing, China.
[3] Zemach, E., D. & Rumisek, A., L. (2003) Success with College Writing: From Paragraph to Essay, Macmillan Languagehouse Ltd., Tokyo.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Understanding Student Answers In Introductory Physics Courses

C.W. LIEW [a,1] JOEL A. SHAPIRO [b] D.E. SMITH [c] and RYAN MCCALL [d]
[a] *Department of Computer Science, Lafayette College*
[b] *Department of Physics and Astronomy, Rutgers University*
[c] *Department of Computer Science, Rutgers University*
[d] *Department of Mathematics, Lafayette College*

Keywords. Mathematics and science education, Physics Algebraic Equations

1. Introduction

Many assigned problems in introductory physics courses require the student to formulate a system of algebraic equations that describe a physical situation. An Intelligent Tutoring System (ITS) designed to help must be able to identify which physical quantity each student variable represents, and which fundamental laws each of her equations incorporates. Many ITS's predefine the variables ([4,2]) or require the student to explicitly and precisely specify what each one represents (*e. g.* ANDES[3]). Our earlier work [1] relaxed this constraint for problems without explicit numerical constants. This paper describes how the technique has been extended to cover problems with numerical constants.

2. Problems and Issues

Determining the meaning of student variables and equations is difficult. To understand why, consider the following problem. A motorcycle cop traveling at 30 m/s is 400 m behind a car traveling at 25 m/s. How long will it take to catch the car? A student might use the relative velocity v_r and give $t = 400\ \mathrm{m}/v_r$, or she might match the positions in time, $d_m = v_m t$, $d_c = 400\ \mathrm{m} + v_c t$, $d_m = d_c$ and solve for t. These inputs need to be matched against a canonical set of variables and equations, which might well not include the relative velocity, and will not include numerical (given) quantities. Our approach to recognizing these variables begins with identification heuristics that list the possible physical concepts normally associated with the letter. Here v_m is likely to be a velocity, and t a time, though in other problems it could be a thickness. These possibilities are constrained by requiring dimensional consistency of all the student equations.

[1] Correspondence to: C.W. Liew,
Department of Computer Science, Lafayette College, Easton PA 18042
E-mail: *liew@cs.lafayette.edu*

Our algorithm attempts to assign numerical quantities used by the student to the canonical variable, so in the examples we would recognize 400 m as the given value of d_0. Numerical values not so identified are then compared to inverses of given values, combinations by addition and subtraction of given values of the same dimension, or the product or quotient of two given values. Had the first example used a numerical value for v_r, by writing $(5\ \mathrm{m/s}) * t = 400\ \mathrm{m}$, the 5 m/s would be found as the difference between v_m and v_c. Having replaced all the constants with the variables they represent, we are left with a purely algebraic system of equations with only dimensionless constants. This is the situation we addressed in our previous paper [1].

3. Evaluation

We evaluated our technique on data from three problems from the ANDES system used at the U. S. Naval Academy in 2001. All involve the use of physics principles from classical mechanics that do not employ trigonometric functions. Exkt4a is essentially the motorcycle cop problem described earlier, and Exkt3a gives the time taken for the shooter to hear a gong he hits with a bullet, given the speed of bullet and sound, and asks how far away the gong was. The last problem is

> *Exe5a*: A frictionless roller-coaster car tops the first hill whose height is 30.0 m above the base of the roller coaster with a speed of 3.00 m/s.
>
> What is the magnitude of the instantaneous velocity at the apex of the next hill which has a height of 10.0m?
>
> Choose the zero of gravitational potential energy at the base of the roller coaster.

The data we extracted from the ANDES logs on these three problems contained only correct equations. While the individual equations are correct, the student submission may be incomplete or contain redundant equations. Because ANDES immediately informs students if their equation is incorrect, and encourages the student to immediately fix it, the available logs did not provide full submissions including wrong equations which were appropriate to include in our test. Our earlier experiments [1] evaluated the ability of the algorithm to detect and analyze incorrect equations to generate useful feedback, but these evaluations were restricted to pure algebraic equations, *i. e.*, they did not contain dimensioned numerical values.

	Exkt3a	%	Exkt4a	%	Exe5a	%
Successful analysis	210	**89.8%**	250	**96.2%**	135	**93.2%**
a) Correct Solution	83	35%	118	45.4%	118	81.4%
b) Partial Solution	110	47%	99	38.1%	11	7.6%
c) No nontrivial equations	17	7.3%	33	12.7%	6	4.1%
Unsuccessful analysis	24	**10.2%**	10	**3.8%**	10	**6.8%**
d) Non-derived equations	6	2.5%	0	0%	4	2.7%
e) Unknown constants	18	7.7%	10	3.8%	6	4.1%
Total	234	100%	260	100%	145	100%

Table 1. Results from ANDES Fall 2001 data

The results of the evaluation are shown in Table 1. Each column shows the number of submissions for each problem in each category and next to it is the percentage. Category

(a) are the submissions that were determined to be correct and complete solutions to the problem. Category (b) are the submissions that contained some number of correct equations but did not contain sufficient equations to solve the problem. Category (c) are the submissions where the student did not enter any equations beyond the the assignment of explicitly given values to variables. Our system successfully handled the submissions in these three categories, which represented 93% of all submissions.

The submissions that could not be fully analyzed fall into two categories. Category (d) are submissions that contain equations that could not be resolved because of limitations with the implemented technique, *e. g.*, common factors have been omitted. Category (e) are submissions that used numerical values that were **not** recognized by the algorithm for a variety of reasons. These reasons are discussed in the next paragraph.

The failure to recognize some numerical constants and certain student equations was due to three main causes: trigonometric functions, detection of implicit zeros, and the presence of common factors. Our algorithm does not handle trigonometric functions due to their numerous equivalent representations (*e. g.*, $\cos(x) = \sin(90 - x)$) and ambiguities in numerical evaluation. We do not handle implicit values of zero that manifest themselves as terms missing from equations (*e. g.*, in a momentum conservation problem with an object initially at rest). Because our method of reducing canonical equations [1] does not include inserting numerical givens, the momentum conservation equation without the zero term is not recognized. Enhancements to handle implicit zeros would not be difficult. Our implementation also fails to recognize equations in which a common factor has been removed, so the equation $g*h_1+0.5*v_1^2 = g*h_2+0.5*v_2^2$ is not recognized as the energy conservation equation $m*g*h_1+0.5*m*v_1^2 = m*g*h_2+0.5*m*v_2^2$.

4. Conclusion and Acknowledgments

We have described how a student's algebraic equations in physics can be analyzed to detect errors and provide useful feedback. Unlike previous approaches, our approach does not require the student to define the variables explicitly (a time consuming, tedious step) and can analyze problems with embedded dimensional numeric constants commonly used in introductory physics courses. The technique has been evaluated on three problems from the ANDES system and is successful over 90% of the time in determining (1) the mapping of the student's variables to specific physical quantities, (2) the correctness of each student equation, and (3) the completeness of the student's submission. We are grateful to Kurt VanLehn, Anders Weinstein and the Andes group for their help

References

[1] Liew, C., Shapiro, J. A., and Smith, D. What is the student referrring to? mapping properties and concepts in students' systems of physics equations. In *Proceedings of the 12th International Conference on Artificial Intelligence in Education* (2005).

[2] `http://www.masteringphysics.com/`.

[3] Shelby, R. N., Schulze, K. G., Treacy, D. J., Wintersgill, M. C., VanLehn, K., and Weinstein, A. An assessment of the Andes tutor. In *Proceeding of the 5th National Physics Education Research Conference* (Rochester, NY, July 2001).

[4] Titus, A. P., Martin, L. W., and Beichner, R. J. Web-based testing in physics education: Methods and opportunities. *Computers in Physics 12, #2* (1998).

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Learning Tutorial Rules Using Classification Based On Associations

Xin LU[1], Barbara DI EUGENIO and Stellan OHLSSON
Department of Computer Science, University of Illinois at Chicago, USA

Abstract. Rules have been showed to be appropriate representations to model tutoring and can be easily applied to intelligent tutoring systems. We applied a machine learning technique, Classification based on Associations, to automatically learn tutorial rules from annotated tutoring dialogues of a human expert tutor. The rules we learn concern the tutor's attitude, the domain concepts to focus on, and the tutor moves. These rules have very good accuracy. They will be incorporated in the feedback generator of an Intelligent Tutoring System.

Keywords. Tutorial rules, Natural language feedback, Expert tutoring

Introduction

To bridge the gap between current Intelligent Tutoring Systems and human tutors, previous studies[1][2] proved that natural language (NL) interfaces could be one of the keys. But it is still not clear what type of NL feedback, and when and how to deliver it in ITSs to engender significantly more learning than simple practice. We are specifically interested in building a computational model of expert tutoring, and to describe how expert tutors will give natural language feedback to their students. In this paper, we present a rule based model of how tutors generate their feedback. This model is motivated by a previous study of ours, in which we found that the expert tutor is indeed significantly more effective than the non-expert tutor[3]. In that study, students were tutored on extrapolating complex letter pattern, such as inferring EFMGHM from ABMCDM. We have already developed a baseline ITS for this task. Our goal is to build a natural language feedback generator for this ITS which will use the tutorial rules we have learned.

Based on the ACT-R theory[4], production rules can be used to realize any cognitive skill. Therefore we can use production rules as a formalism to computationally model expert tutoring. The rules can be designed manually or learned from the human tutoring transcripts. The dialogue management of AUTOTUTOR[5] embeds a set of 15 fuzzy production rules to select the next dialogue move for the tutoring system. The CIRCSIM group has applied machine learning to discover how human tutors make decisions based on the student model[6]. They used Quinlan's C4.5 decision tree learning algorithm[7] to find tutoring rules. They obtained about

[1] Correspondence to: Xin Lu, Computer Science (M/C 152), University of Illinois at Chicago, 851 S Morgan St., Chicago IL, 60607, USA. Email: xlu4@uic.edu

80%~88% accuracy only within the training data set which is also an extremely small sample. Since they only reported accuracy on training but not on testing, it's very hard to understand how good their approach is.

1. Method

Classification based on associations (CBA) [8] which integrates classification and association rule mining can generate class association rules and can do classification more accurately than C4.5. Classification association rules (CARs) are association rules with the target on the right hand side of the rules. A CAR is an implication of the form: $X \rightarrow y$, *where* $X \subseteq I$, *and* $y \in Y$. X is a set of features. I is the set of all features. y is the target class. Y is the set of all classes. CBA also provides strength measurements for the CARs:

- **Support:** The rule holds with support *sup* if *sup*% of cases contain X or y.
- **Confidence:** The rule holds with confidence *conf* if *conf*% of cases that contain X also contain y.

So when CBA does classification, more than one rule can fit a certain case and the final class will be derived from the rule with highest confidence. If the confidence of the rules that apply is the same, the rule with highest support will be picked. Again if the support is also equal, CBA will classify the case according to the rule which is generated earlier than the others. Of course, there will be some cases that no CARs can classify the case. CBA saves a default class to deal with this kind of situation.

2. Experiments and Results

We collected tutoring dialogues in tutoring the letter pattern extrapolation task with three tutors, one expert and two non-experts. All the sessions are video recorded and 12 dialogue excerpts were transcribed from 6 different subjects with an expert tutor solving two problems in the curriculum. On the transcript we annotated tutor move, tutor's attitude, student move, correctness of student move, student action, student input, student's confidence, hesitation time, the letter relationship currently focused on, and the relationship scope within the problem.

Features used in the rules are the annotations of the tutoring dialogues and the student's knowledge state on each type of letter relationship, which is computed from other annotations within each dialogue excerpts. CBA will automatically generate rules with a subset of these features. Tutorial dialogues are time series data, which means that the prediction of what the expert tutor should do now should be based on information of the last few utterances. In our experiments, we used the features from only the last utterance.

Using 12 dialogues with 6-way cross validation, we did 4 experiments to learn tutorial rules for choosing the tutor's attitude, the letter relationship which the tutor will talk about, the relationship scope within the problem which the tutor will focus on, and the tutor move. Tutor's feedback to students can be divided into 3 categories according to the tutor's attitude towards students: positive, neutral and negative. The letter relationship is the basic concept in the letter pattern task. The relationship scope

Table 1. Experiment Results

Prediction Category	**Accuracy in Training**	**Accuracy in Testing**
Tutor's attitude	93.628%	89.490%
Letter relationship	95.987%	90.035%
Relationship scope	95.097%	88.421%
Tutor move	78.261%	56.789%

concerns the coverage of each type of letter relationship. During tutoring, tutors need to choose the concepts to teach students and discuss with them, and also need to decide how to break down the problem and choose an appropriate coverage. A tutor move is akin a response strategy.

Table 1 reports accuracy on training and testing in learning these four sets of tutorial rules. The accuracy for tutor moves is not as high (only about 78% in the training data and 57% in the testing data). One of the possible reasons is that among 8 categories of tutor move, three (summarizing, prompting, instructing) are very difficult to distinguish, even for human annotators. For example, we found that the two annotators disagreed a lot as regards "summarizing" and "instructing". However, the results are sufficient as a basis for our experiments, since our ultimate evaluation measure is whether the natural language feedback generated based on these rules can improve learning.

Acknowledgements

This work was supported by grant N00014-00-1-0640 from the Office of Naval Research, and partly, by awards IIS0133123 and ALT 0536968 from the National Science Foundation.

References

[1]B. Fox. The Human Tutorial Dialogue Project. *Lawrence Erlbaum Associates, Hillsdale*, NJ, 1993.
[2]B. Di Eugenio, D. Fossati, D. Yu, S. Haller and M. Glass. Aggregation Improves Learning: Experiments In Natural Language Generation For Intelligent Tutoring Systems. *The 42nd Meeting of the Association for Computational Linguistics*, Ann Arbor, MI, 2005.
[3]B. Di Eugenio, T. C. Kershaw, X. Lu, A. Corrigan-Halpern, and S. Ohlsson: Toward a computational model of expert tutoring: a first report. In *Proceedings of 19th International conference of Florida Artificial Intelligence Research Society*, 2006, 503-508.
[4]J. R. Anderson, C. F. Boyle, A. T. Corbett, and M. W. Lewis: Cognitive modeling and intelligent tutoring. *Artificial Intelligence*, 42(1), 1990.
[5]A. Graesser, N. K. Person, and D. Harter: Teaching tactics and dialog in autotutor. *International Journal of Artificial Intelligence in Education*, 2005.
[6]Y. Zhou: Building A New Student Model to Support Adaptive Tutoring in a Natural Language Dialogue System. Doctoral dissertation, Illinois Institute of Technology, 2000.
[7]R. J. Quinlan: C4.5: Programs for Machine Learning, 1993.
[8]B. Liu, W. Hsu, and Y. Ma: Integrating classification and association rule mining. In *Knowledge Discovery and Data Mining*, 1998, 80–86.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

The Effect of Problem Templates on Learning in Intelligent Tutoring Systems

Moffat MATHEWS[1] and Antonija MITROVIC[2]
Intelligent Computer Tutoring Group, University of Canterbury, Christchurch, New Zealand
Email: {moffat[1], tanja[2] }@cosc.canterbury.ac.nz

Abstract: This paper proposes the notion of *problem templates* (PTs), a concept based on theories of memory and expertise. These mental constructs allow experts to quickly recognise problem states, almost instantaneously retrieve potentially vast amounts of domain-specific information, and seemingly effortlessly implement strategies to solve the problem. We investigate the role of problem templates in intelligent tutoring systems. An evaluation study was conducted in which PTs were created and implemented in SQL-Tutor. PTs were used to model students and make pedagogical decisions. Students using templates showed high levels of learning within short periods of time. Further evaluation studies are required to investigate the extent and detail of its effect on learning.

Introduction

The main goal of Intelligent Tutoring Systems (ITS) is to increase the effectiveness of learning. In this paper, we present the notion of Problem Templates (PTs), or high-level mental constructs used by experts to hold large amounts of relevant domain-specific information that is retrievable as a single chunk. Our goals were to examine the validity of such a construct, investigate if physical representations of PTs could be created, explore if pedagogical decisions within an ITS could be based on them, and if students' domain knowledge could be modelled using them.

Problem templates, as proposed in this paper, are chunks of domain-specific knowledge, compiled mentally by experts, used to solve commonly occurring problems in a particular domain. PTs are an extension to memory templates as proposed by the Template Theory (TT) [1]. Within PT slots, experts hold information to *recognise particular problem states, common strategies to solve* the problem (i.e. take the user from the current state to an intended solution state), and if required, a list of *tools and associated information* (or pointers to associated templates) required to implement these solutions. Because of PTs, experts have access to vast amounts of domain-specific information applicable to the current context. They are able to almost instantaneously recognise problem states within their domain, and seemingly effortlessly implement solutions. Comparisons of expertise traits [2] and various theories of memory [3] support the existence of PTs. In the driving instruction domain, students are taught templates (hill starts, 3-point turns etc) compiled by experts to increase their rate of learning. Experts in domains (e.g. chess [1]) compile mental libraries of templates. Templates are even used collaboratively in *game plays* to solve problem states between players in team sports such as hockey or football.

1. Introducing PTs to SQL-Tutor

We developed problem templates for SQL-Tutor [4], by categorising the 278 existing problems into 38 groups, whereby each group contained problems with the same solution structure. All relation and attribute names in solutions were then replaced by variables, leaving the generic SQL statements representing common solution strategies. We also developed a feedback message for each template, which is shown to the student on violation of the template. See Table 1 for two examples of SQL templates. As can be seen, template 1 is relevant for ten problems (whose ids are listed). The template also contains the general form of the goal SQL statement (third component), and finally the feedback message.

Table 1: Representation of SQL PT 1 and PT3

(1 (1 26 59 132 135 164 168 151 199 235) ("SELECT * FROM table" "Retrieve all attributes of one table”))
(3 (152 237 255 260) ("SELECT DISTINCT attribute(s) FROM table" "You want the details without duplicates (DISTINCT) of the attribute(s) of a table"))

When selecting a new problem, the control group students (using the original version of SQL-Tutor) were first presented with a choice of SQL clauses. To maintain the similar number of options for the experimental group, we generalised the 38 PTs into eight template groups, which were shown to students. SQL-Tutor highlighted one option as being preferred, but allowed the student to select a different one. In the latter case, the student was presented with a confirmation dialogue, alerting them to their deviation from the preferred choice. The student was then presented with the set of problems appropriate to their selection and their knowledge level. In this set, the preferred problem choice was highlighted; this choice could also be overridden by the user. After selecting the problem, the problem text with solution area was presented to the user. In both versions, the student could submit the solution at any stage. In addition to the feedback based on constraints, participants in the experimental group also received feedback associated with templates.

In the original version of SQL-Tutor, student models are represented as overlays on top of constraints; the experimental version used a similar process but with templates. Although constraint histories were recorded in student models for both groups for comparative analysis, they were used in pedagogical decisions only for the control version; template histories were used for the experimental group. Problem selection was based on the problem difficulty level (or template difficulty level), the student’s knowledge scores, and the student’s knowledge level.

2. Evaluation

An evaluation study was conducted at the University of Canterbury in May 2006 with volunteers from an undergraduate database course, who used SQL-Tutor either in the weekly lab sessions or at their leisure. Out of 68 students enrolled in the course, 48

logged into SQL-Tutor and completed the pre-test. Participants were randomly allocated into two groups: the control (23) and the experimental (25). Pre-tests results showed no significant differences between the two groups: 42% (6%) and 50% (7%) for the experimental and control group respectively.

Unfortunately, only ten students completed the post-test, and consequently the data is insufficient for in depth comparisons. However, the experimental group students needed significantly shorter time (58% of the time needed by the control group students) to solve the same number of problems (p=.055), with fewer attempts. Learning curves for the two groups are very similar, and closely fit the power curves: $y=0.0995x^{-0.384}$ for the control group (R^2=0.92), and $y=0.1044x^{-0.4079}$ for the experimental group (R^2=0.89). This indicates that both groups learned domain-specific knowledge at similarly high rates, and did so equally well.

Some restrictions to this research exist. First, the study was relatively short (four weeks). Generally, under non-ITS instruction (e.g. sports), students take much longer periods of time to compile (create and organise) PTs. Compiling templates however, takes them from being intermediate users to mastery. Second, PTs taught by human tutors are generally compiled by combining the knowledge of many experts and refined over a period of years, while PTs used in this study were compiled by a single person. Third, although PTs can be learned implicitly and automatically, they are generally taught explicitly. In this study, students did not undergo formal teaching of PTs. In spite of these restrictions, the experimental group learned domain-specific knowledge at an equally high rate, and attempted and solved the same number of problems in shorter periods than their counterparts in the control group.

The goals of this research were to examine the validity of PTs, explore if physical representations of PTs could be created and implemented in an ITS, investigate the use of PTs in student modelling and pedagogical decisions, and examine its effects on learning. Relevant prior research and real-life examples support the notion of PTs. Physical representations of problem templates were created in a complex, rich, and relatively open-ended domain (SQL). Problem templates were used for student modelling, problem selection and feedback generation. Lastly, an evaluation study was conducted in which participants used either a control version of SQL-Tutor (using only constraints), or the experimental version (using templates). Preliminary results were also promising, showing equally high rates of learning in both groups.

Future work includes conducting an evaluation study over a longer period of time to collate sufficient data, exploring applicability of PTSs to other instructional domains, and conducting scientific studies to comprehend the processes used by human tutors and students when using PTs.

References

[1] F. Gobet and H. A. Simon, Templates in Chess Memory: A Mechanism for Recalling Several Boards, Cognitive Psychology, 31, 1-40, 1996.

[2] J. Shanteau, The Psychology of Experts: An Alternative View, in Expertise and Decision Support, G. Wright and F. Bolger, Eds. NY: Plenum Press, 11-23, 1992.

[3] F. Gobet, Expert Memory: A Comparison of Four Theories, Cognition, 66, 115-152, 1998.

[4] A. Mitrović, An Intelligent SQL Tutor on the Web, Artificial Intelligence in Education, 13, 173-197, 2003.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Situation Creator: A Pedagogical Agent Creating Learning Opportunities

Yongwu MIAO [a], Ulrich HOPPE [b] and Niels PINKWART [c]
[a] *Educational Technology Expertise Center, Open University of the Netherland,*
[b] *Institute of Computer Science and Interactive Systems, University of Duisburg-Essen*
[c] *Computer Science Institute, Clausthal University of Technology*

Abstract. In a multi-user, real-time, and situation-based learning environment, the availability of enough and appropriate situations is crucial for success. In order to improve effectiveness and efficiency of learning, we develop a new type of pedagogical agent: situation creator. Such an agent intentionally creates specific situations in the shared virtual driving place according to users' performance information. We conduct a pilot evaluation and found that the situation creators significantly increase the number of situations that a learner can expect to encounter while using the system.

Introduction

In a typical training course with a simulator, an individual trainee interacts with the simulation system through a series of pre-defined situations. We adopted an alternative design approach to a car-driving simulation environment for learning. Rather than a single user going through a series of pre-defined driving scenarios, in our learning environment geographically distributed users can learn to handle various situations by virtually driving in a shared driving place in a way analogous to driving in the real world. If a user needs help or can not behavior correctly, an *coach agent* will provide a specific guidance to the user based on the user's experience concerning the current situation [1]. In addition, in order to increase the volume of traffic and traffic situations, we introduced *driver agents*, which head for random destinations on the roads and never consider the needs of learners. The problem of using randomly acting driver agents is that they are not highly effective in situation generation, although they make the virtual driving place more crowded.

In order to improve effectiveness and efficiency of learning, we develop a concept of *situation creator* [2]. Such an pedagogical agent intentionally creates specific situations in the shared virtual driving place according to users' performance information. In this paper, we present the design and implementation of a situation creator example and report the result of a pilot study with the system.

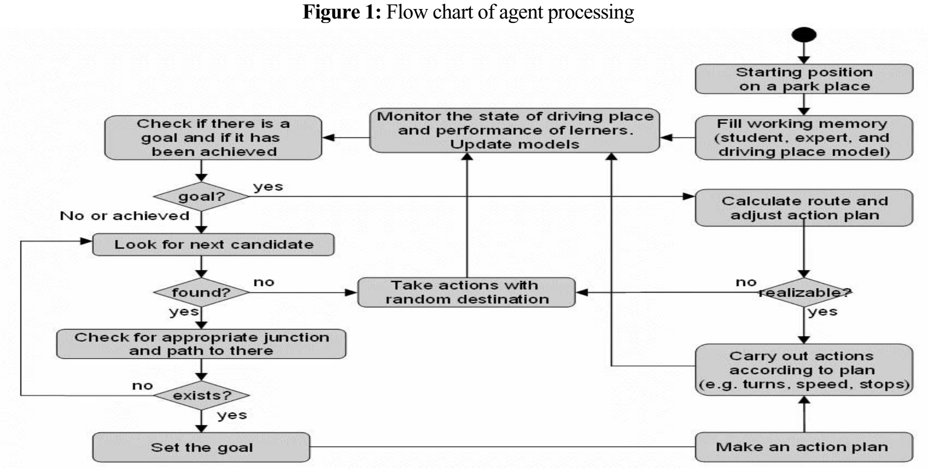
1. Design and Implementation of an Situation Creator

Figure 1 shows the algorithm for the situation creator example. Once it becomes active, it starts to automatically navigate from a parking place to the shared driving place where the learners practice driving. The situation creator then seeks cars controlled by learners. As long as there are learners in the shared driving place, the situation creator will try to create a learning opportunity for one of them. The agent first selects its "target" learner as the one whose car is closest to his own. It looks up in the learner model in which situations the

candidate needs training according to his/her past performance. If a realizable situation for the candidate can be determined, it means that the agent has a goal. Otherwise, the agent tries to consider the next candidate. Whether a situation is realizable or not depends on the current state of the learner's car (e.g., direction and velocity), the driving place (e.g., road network and an adequate location for creating the situation), and the state of the agent's car. For example, when considering a situation "junction without right of way", the situation creator looks for a junction on the prospective way of the learner. The junction has to fulfill the condition that the learner has to give the right of way to crossing cars. Thus it looks for a yield or stop sign. If there is such a junction on the way that the learner's car is driving, the agent checks if there is an appropriate route for himself to reach this junction just a little earlier than the learner. If all conditions can be met, the goal is determined.

Once the agent has determined a realizable goal, it tries to create the target situation. A goal is relatively stable. It maintains unchanged for a period of time until it is achieved or turns out to be unrealizable. To check this, the agent contineously monitors the road map and the states of the cars. E.g., if the learner does not head for the "target junction" anymore, or the agent is delayed for any reason so that the learner is expected to pass the junction before the agent can arrive, the situation creation is aborted and a new search will start. If the state of the environment changes unexpectedly (e.g., the learner's car slows down and is expected to arrive later) but the situation is still realizable, the agent can adjust its action plan (e.g., it slows down or even stops and to wait for the learner to arrive) and still create the target situation.

Figure 1: Flow chart of agent processing



2. Methods

In order to verify whether our approach of using situation creators to create learning opportunities for learners really works as intended, we have conducted a small pilot study. Since the effectiveness of the whole approach essentially relies on the prerequisite of agents being able to actually create the learning opportunities, we measured how many learning opportunities the situation creators could create, compared to the driver agents.

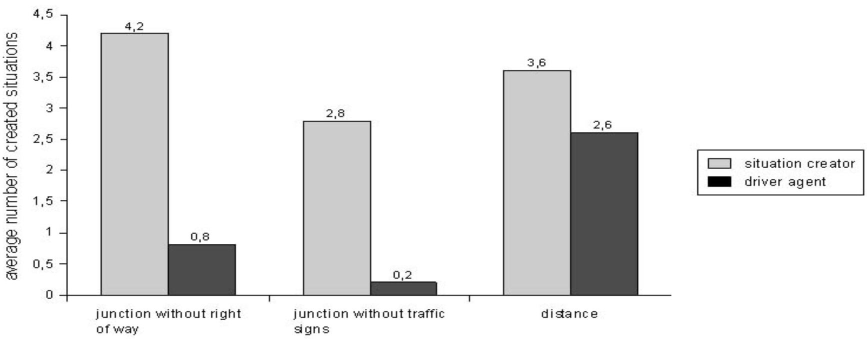
In our pilot study, the system has been used by 5 groups of 2 learners each. Each group used the system for two sessions of 10 minutes. In the first session we asked the groups to drive together with driver agents. In the second session we replaced these driver

agents with situation creators. In each session the two learners drove together with five agents. To analyze the results of the two sessions, the number of occurred situations was counted. In addition to this, we used agent log files to track how often a situation creator tried to create a situation for a particular learner, and whether it was successful in this or not. For our evaluation, we focused on three situations: “junction without right of way”, “junction without traffic signs”, and “safety distance to another car”.

3. Results

Our data analysis shows two immediate results. First, the data confirms that without agents learners rarely encounter the three situations (of course, this heavily depends on group size and driving place size). The average number of situations created by learners was 0.8 for the junction situations and 1.7 for the safety distance situation. Both values are rather low considering the driving time of 10 minutes. Our second finding is that situation creators can solve this problem and that driver agents are not sufficient for creating complex situations. This is illustrated in figure 2, which shows the average number of situations created by driver agents and situation creators in the test sessions. Situation creators generated significantly more situations of all three types than the learners created themselves (t-tests, $p<0.01$ for all three situations). Also, situation creators generated significantly more situations of the two “junctions” types than driver agents ($p<0.0001$). In summary, the results of our evaluation show that the situation creators are effective and superior to driver agents.

Figure 2: Situation creation by agents



4. Conclusions

We have presented the design and imeplementation of a simple situation creator as an example that creates situations involving the right of way rules at traffic junctions. We conducted a pilot study. The evaluation results indeed confirmed that the situation creators generated much more learning situations for a learner than the random driver agents, and significantly increased the number of situations that a learner can expect to encounter.

References

[1] Miao, Y., Pinkwart, N., and Hoppe, U. (2006) A Collaborative Virtual Environment for Situated Learning of Car Driving. *Advanced Technology for Learning Journal*, 3(4), 233-240, ACTA Press.

[2] Miao, Y., Hoppe, U. Pinkwart, N., Schilbach, O., Zill, S., and Schloesser, T. (2006) Using Agents to Create Learning Opportunities in a Collaborative Learning Environment. *Proceedings of the 8th international conference on Intelligent Tutoring Systems.* p 798-800, Berlin, Springer.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Fitting Spatial Ability into Intelligent Tutoring Systems Development

Nancy MILIK[1], Antonija MITROVIC[1] and Michael GRIMLEY[2]
[1]Department of Computer Science and Software Engineering
[2]Department of Education
University of Canterbury, New Zealand

Abstract. Building effective learning environments is an art that can only be perfected by a great deal of explorations involving the environments' audience: the learners. This paper focuses on taking into account the learners' spatial ability into the development of Intelligent Tutoring Systems. We modified ERM-Tutor, a constraint-based tutor that teaches logical database design, to provide not only textual feedback messages, but also messages containing combinations of text and pictures, in accordance with the multimedia theory of learning [1]. Results of a preliminary study performed show a promising indication for further explorations. We plan to use these results as the basis for another evaluation study in early 2007.

Introduction

Intelligent Tutoring Systems (ITSs) are effective learning tools due to the adaptive pedagogical assistance they provide. Students differ in their capabilities for learning and processing information. This paper describes a project which focuses on spatial ability, a psychometric construct essential to activities related to spatial reasoning, such as the ability to manipulate images or spatial patterns into other arrangements [2]. Learners with high spatial abilities perform better with graphic or spatially-oriented content than those with low spatial ability. It is worth noting, however, that a low spatial ability score is not a deficit; there is evidence that it can be improved through training and practice [3]. Nevertheless, enhancing ITSs to accommodate low spatial ability learners could be beneficial for their problem solving skills. As a consequence, learners with different spatial abilities should receive different types of content.

The theory of multimedia learning states that "*[multimedia] design effects are stronger for low-knowledge learners than for high-knowledge learners and for high spatial learners rather than from low spatial learners*" (p. 161) [1]. Low-spatial learners must devote much of their cognitive capacity to process multimedia information. High-knowledge and low-spatial learners are able to use their prior knowledge to compensate for the cognitive load needed to integrate the information received by the dual-channel. Therefore, it is the combination of the learners' spatial ability and level of knowledge that influences their meaningful/deep learning.

We present an approach to support the learners' spatial ability in ERM-Tutor [4], a constraint-based ITS that teaches logical database design (i.e. the algorithm for mapping conceptual to logical database schemas). The next section presents the modifications made in this project. We then describe the preliminary study and the results obtained, followed by conclusions and future work in the final section.

1. Spatial Ability Support in ERM-Tutor

Influenced by Mayer's work, we created a new version of the system. The original ERM-Tutor provides only text-based feedback. Following the multimedia learning theory, we decided to incorporate a pictorial aspect in the messages; for each feedback message, we created a graphically annotated (multimedia) version.

Each feedback message in ERM-Tutor is associated with a constraint. In other words, each constraint has a feedback message which is returned when the constraint is violated. Consequently, each message provides a hint on how to satisfy its particular constraint. To make the original and the newly created messages comparable, we kept the text identical in both versions. The only difference is the addition of a pictorial representation in the new version. Figure 1 shows an example message in multimedia representation. A total of 112 images were created, each corresponding to a single feedback message. In addition, ERM-Tutor was modified to cater for both versions of feedback and prepared for an evaluation study described in the following section.

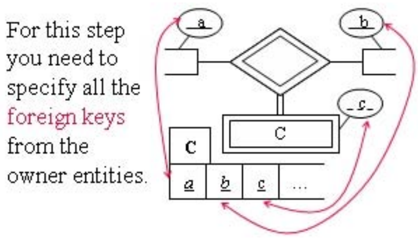


Figure 1. An example feedback message in multimedia representation

We also explored two cognitive tests for testing spatial ability [5]: a ten-item Paper Folding Test intended to evaluate a component of spatial ability called visualization, and an eighty-item mental Card Rotation Test which evaluates spatial orientation. Each test has a three-minute time limit and is suitable for ages 13-18.

2. Experiment

We preformed a preliminary study with students enrolled in an introductory database course at Canterbury University in March 2006. Our hypothesis is that students with a high spatial ability level will benefit more from multimedia feedback than students with a low spatial ability, given the same background knowledge. As each student's spatial ability level (either high or low, as opposed to the actual value) is determined relatively to the sample group, it was decided to compute it post-hoc. The students were randomly allocated to one version of the system, providing either textual or multimedia feedback. The assumption was that each group would ultimately include students with high and low spatial abilities. Therefore, the experiment allows for a 2x2 comparison: textual messages for high (TH) and low spatial ability students (TL), and multimedia messages for high (MH) and low spatial ability students (ML).

The study was conducted in two two-hour sessions of scheduled labs on ER mapping, straight after students had attended lectures on the topic. Each participant attended one of the sessions, and worked with ERM-Tutor individually, solving problems at his/her own pace. The pre-test was collected at the start of the session, while the post-test was administered after two hours of interaction.

55 students participated and completed both spatial tests. The test score for the paper fold test was 6.89 out of a possible 10. The total possible score for the card rotation test is 80. We computed the total test score by dividing the score by 8, thus giving a range of 1–10. The students scored a mean of 6.43. To compute the spatial

ability of each student, we added both test scores giving a possible range of 1–20. Using a median split, a total of 28 students scored above the median and were classified as high spatial, and the other 27 students were classified as low spatial.

The system recorded all student actions in logs. Due to a technical problem however, the logs from the first session could not be used. A total of 17 students used the system for more than 10 minutes. On average, students attempted 3.4 problems and completed 33% of them.

Only 13 students completed both tests, scoring a mean of 1.92 (sd = 1.04) on the pre-test, and 3 (sd = 1.15) on the post-test, resulting in significant improvement of their performance ($t=3.09$, $p < 0.001$). The scores for the four groups are given in Table 1. These preliminary results (although with small numbers) seem to refute Mayer's prediction that high spatial learners will benefit most from multimedia messages. However, it seems that the participants from the TH and MH groups who completed both tests started with higher pre-existing knowledge, and therefore the theory may be more pertinent in that low knowledge individuals will have a higher gain.

The numbers of valid logs in each condition are too small, and we are therefore unable to closely analyze the effect of the students' spatial ability on their performance. Analyses of the questionnaires showed that students who received multimedia feedback rated the overall quality of the feedback messages 25% higher (mean of 4 out of a possible 5) than those who received textual feedback (mean of 3 out of a possible 5).

Table 1. Pre/post test results for the students who sat both tests

	Low spatial			High spatial		
Feedback	No.	Pre-test	Post-test	No.	Pre-test	Post-test
Textual	TL: 4	1.5 (1)	3.5 (0.6)	TH: 3	2 (1)	2.3 (0.6)
Multimedia	ML: 2	1.5 (0.7)	3.5 (0.7)	MH: 4	2.5 (1.3)	2.75 (1.9)

3. Conclusions

This paper has looked at the potential of accounting for spatial ability in ERM-Tutor. Results from a preliminary study show an overall improvement in the students' domain knowledge level after two hours of interaction. We could not however report any findings on the correlation between spatial ability, content representation and the learning experience due to a technical problem. Although the amount of data collected was small, the results show a promising indication for further explorations. We plan to use this study as the basis for another evaluation study testing the same hypothesis.

References

1. Mayer, R., *Multi-media Learning*. 2001: University of California.
2. Carroll, J., *Human cognitive abilities*. 1993, England: Cambridge University Press.
3. Baenninger, M. and N. Newcombe, *The role of experience in spatial test performance: A meta-analysis*. Sex Roles, 1989. 20(5): p. 327-344.
4. Milik, N., M. Marshall, and A. Mitrovic. *Responding to Free-form Student Questions in ERM-Tutor*. in *Proceedings of 8th International Conference on Intelligent Tutoring Systems*. 2006. Taiwan: Springer.
5. Ekstrom, R., J. French, and H. Harman, *Manual for kit of factor referenced cognitive tests*. 1997: Princeton.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Further Evidence for the Deep-Level Reasoning Question Effect

Brent MORGAN, Roby COLES, Josh BRITTINGHAM and Barry GHOLSON
The University of Memphis
202 Psychology Building, University of Memphis, Memphis, TN, 38152, USA

Abstract. This experiment sought to generalize to middle- and high-school students the learning gains observed through deep-level reasoning questions among college students in vicarious environments. 342 eighth through eleventh graders were presented with either content alone, content preceded by deep-level reasoning questions, or an interactive session with a virtual tutor. Students exposed to the deep-level reasoning questions outperformed the interactive and content-only groups, suggesting that deep-level reasoning questions do indeed aid in knowledge representation amongst middle- and high-school students.

Keywords: vicarious learning, intelligent tutoring

Introduction

Recent advances in computer-based courses [1] have created situations in which learners frequently find themselves in settings in which they are observers [2] rather than active participants in the learning process. These educational advances represent a challenge to researchers in further understanding the conditions that support knowledge acquisition in education settings in which learners are relatively isolated. To extend this line of inquiry, several avenues have been pursued focusing on vicarious learning environments. In these environments, students hear and/or see the addressees but have no way of interacting with the source of the content they are attempting to master.

It has long been known that question generation is one of the processing components that supports comprehension and problem solving [3]. Previous research has shown that college students who receive deep-level reasoning questions prior to content statements outperform students who receive shallow questions or none at all on a multiple choice posttest [4] and write significantly more content in a free-recall exercise [5]. Furthermore, when allowed to generate their own questions, students exposed to deep-level questions, generated more questions exhibiting deep reasoning than those who received shallow questions [5].

The issue addressed in the current research is this: how can computer-based instruction be designed to support knowledge acquisition processes when learners from middle school and high school cannot physically interact with or control the content they are attempting to master? The present study investigated the relationship between deep-level reasoning questions and learning on the topics of Newtonian physics and computer literacy.

1. Methods

The participants were 8th-11th graders in the Memphis City School system. Students were randomly assigned to one of three learning conditions: interactive, monologue-like, and dialogue. In the interactive condition, participants interacted directly with AutoTutor (a computerized agent with facial expressions and some gesturing which serves as a conversational partner with the learner). In the monologue condition, participants were presented with the sentences included in the ideal answer and the expectations of each topic. In the dialogue condition, a deep-level reasoning question preceded each sentence in the ideal answer and the expectations of each topic. Table 1 presents the number of participants at each grade level in each of the three conditions.

Table 1

Grade	Dialogue	Interactive	Monologue	Total
8th	27	18	29	**74**
9th	30	34	27	**91**
10th	30	30	30	**90**
11th	27	30	30	**87**
Total	**114**	**112**	**116**	**342**

Participants first completed the Gates-McGinite reading test (35 minutes) and a pretest that included 26 questions for the Newtonian physics condition [6] and 24 questions for the computer literacy condition [4, 7]. There were two tests in each domain of the same length that were counterbalanced as pretests and posttests across participants in each group. After completing the pretest, participants were exposed to one of the three learning conditions. The computerized sessions were 37 minutes in length, after which they were subsequently administered the posttest. Altogether the complete study required parts of three classroom sessions (about 130 minutes) for each participant who was tested in the schools and about 130 minutes for those who were tested in a laboratory setting at the University. Each session in the schools included about 22 students, while the laboratory settings included about eleven students in each.

2. Results and Discussion

Planned comparisons on pretest to posttest scores showed that students in the dialogue condition outperformed those in the interactive and monologue conditions, $F(1,334) = 5.19$, $p < .05$. This provides support for what Craig et al. [4] called the "deep-level reasoning questions effect" among eighth to eleventh graders in both the domains of computer literacy and Newtonian physics. The effect had previously been shown among college students in the domain of computer literacy [4, 8].

To date, our research on dialogue and question asking has involved the domains of computer literacy and Newtonian Physics. Thus, it is necessary to explore the deep-level reasoning questions effect in other domains in future research. Second, both the questions and the course content that followed each were spoken by voice engines earlier and in the current research. Would the same learning gains be achieved if the deep-level reasoning questions were presented as on-screen printed text and embedded

in course content that is also presented as on-screen printed text? Third, are all deep-level reasoning question category frames equal when it comes to supporting knowledge acquisition processes on the part of vicarious learners during dialogue? We simply drew our deep-level reasoning questions frames unsystematically from six categories in a question taxonomy presented by Graesser and Person [9] (pp. 110-111). The deep-level reasoning question frames were comparison, interpretation, causal antecedent, causal consequent, instrumental/procedural, and enablement. So the question is: what specific features of which categories of deep-level reasoning questions during dialogue best support knowledge acquisition processes during vicarious learning? Finally, once those features are identified, exactly how do they support cognitive learning processes, and what other kinds of dialogue moves involve these same features?

Acknowledgements

This research was supported by Institute of Education Sciences (R305H050169). Any opinions, findings, conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of IES.

References

[1] Anderson, J. R., Corbett, A. T., Koedinger, K. R., & Pelletier, R. (1995). Cognitive tutors: Lessons learned. *Journal of the Learning Sciences, 4*, 167-207.

[2] Schober, M. F., & Clark, H. H. (1989). Understanding by addressees and overhearers. *Cognitive Psychology, 21*, 211-232.

[3] Graesser, A. C., Baggett, W., & Williams, K. (1996). Question-driven explanatory reasoning. *Applied Cognitive Psychology, 10*, S17-S32.

[4] Craig, S.D., Sullins, J.S., Witherspoon, A.M., & Gholson, J.B. (2006). The Deep-Level-Reasoning-Question Effect: The Role of Dialogue and Deep-level Reasoning Questions During Vicarious Learning. *Cognition and Instruction, 24*, 565-591.

[5] Craig, S. D., Gholson, B., Ventura, M., Graesser, A. C., & the Tutoring Research Group. (2000). Overhearing dialogueues and monologues in virtual tutoring sessions: Effects on questioning and vicarious learning. *International Journal of Artificial Intelligence in Education, 11*, 242-253.

[6] Hestenes, G., Wells, M., & Swacjgannerm G, (1992). Force concept inventory. *The Physics Teacher, 30*, 141-158.

[7] Craig, S. D., Driscoll, D., & Gholson, B. (2004). Constructing knowledge from dialogue in an intelligent tutoring system: Interactive learning, vicarious learning, and pedagogical agents. *Journal of Educational Multimedia and Hypermedia, 13*, 163-183.

[8] VanLehn, K., Graesser, A. C., Jackson, G. T., Jordan, P., Olney, A., & Rosé, C. P. (in press). Natural language tutoring: A comparison of human tutors, computer tutors and text. *Cognitive Science*.

[9] Graesser, A. C, & Person, N. (1994). Question asking during tutoring. *American Educational Research Journal, 31*, 104-137.

A Framework for Problem-Solving Knowledge Mining from Users' Actions

Roger NKAMBOU[1], Engelbert Mephu NGUIFO[2] ,
Olivier COUTURIER[2] and Philippe FOURNIER-VIGER[1]
[1]*Université du Québec à Montréal (Canada),* [2]*CRIL-CNRS, IUT de Lens (France),*
nkambou.roger@uqam.ca, mephu@cril.univ-artois.fr

Abstract. In this paper, we present an approach to capture problem-solving knowledge using a promising technique of data and knowledge discovery based on a combination of sequential pattern mining and association rules discovery.

Introduction

This paper proposes an approach to support new domain knowledge discovery in procedural domain where it is difficult to set up a clear problem space or task models. In such a domain, we need to capture new procedures (correct or incorrect), new problem spaces and new problem-solving strategies from users' actions. We know how time consuming is the cognitive task analysis that aims at producing effective problem space or task model to support model and knowledge tracing, coaching, errors detection and plan recognition. How can we build this complex structure by learning from users' interactions with an intelligent tutoring system (ITS)?

This paper presents an approach based on a combination of sequential pattern recognition and association rule discovery. We show how the proposed framework is used to discover new knowledge in a given domain which then extends domain knowledge and serves as a problem space allowing the intelligent tutor to track learners' actions and give relevant hints when needed.

1. Problem-solvingProblem-Solving Data Representation

In cognitive tutors, problem-solving knowledge is represented as procedures each corresponding to a possible path to a successful or unsuccessful solution to the problem. A procedure (or a plan) is usually represented as a sequence of atomic actions. Actions are events that occur at a given time. Table 1 shows a dataset of 8 successful plans (captured from a real usage of CanadarmTutor [4]) where each entry corresponds to a plan's events. From this dataset, it is possible to easily compute frequent sequences of actions using a minimal support (*minsup*) defined by the user. A sequence is said to be frequent if it occurs more than *minsup* times.

PlanID	Sequences of actions
P1	1 2 25 46 48 {9 10 11 31}
P2	1 25 46 54 79 {10 11 25 27}
P3	1 2 3 {9 10 11 31} 48
P4	2 3 25 46 11 {14 15 16 48} 74
P5	2 25 46 47 48 49 {8 9 10}
P6	1 2 3 4 5 6 7
P7	25 26 27 28 30 {32 33 34 35 36}
P8	46 54 76 {10 27} {48 74}

Table 1: A data set of 8 successful plans.

Sequential pattern	Sequence pattern label
1 25 46 48	S1
1 25 46 {10 11}	S2
1 {9 10 31}	S6
1 {9 11 31}	S7
1 {9 10 11 31}	S8
1 46 {10 11}	S13

Table 2: Examples of sequential patterns extracted by PrefixSpan with their associated labels

2. The Proposed Framework

The proposed framework integrates several stages including three important processes (in bold). Its main scheme is as follow:

Log files containing users plans → Automatic coding of data → ***Sequential Patterns Finding (PrefixSpan)*** *→ Frequent patterns →* ***Building of the Meta-Context*** *→ Meta-Context →* ***Association Rules Finding (IGB)*** *→ New procedural task knowledge (PTK)*

Finding Frequent Sequential Patterns Using PrefixSpan. We choose the PrefixSpan approach [1] as it is one of the most promising ones for mining large sequence databases having numerous patterns and/or long patterns, and also because it can be extended to mine sequential patterns with user-specified constraints. PrefixSpan is a projection-based, sequential pattern-growth approach that recursively projects a sequence database into a set of smaller projected databases. Sequential patterns are grown in each projected database by exploring only locally frequent fragments. Table 2 shows examples of sequential patterns extracted by PrefixSpan from data in table 1 using a minimum support of 25%.

Extracting Generic rules Between Sequences Using IGB. Links between sequential patterns can lead to some interesting domain rules. In our case, we are interested by minimal and non redundant association rules between patterns, also called generic bases. Among previous studies on mining of generic bases, we choose IGB [2] which takes as input the meta-context of initial plans, and two thresholds which are the minimum support, *minsup* (already defined in PrefixSpan), and the minimum confidence, *minconf*. The meta-context of initial plans (see example in table 3) is the set of plans redefined with the frequent sequential patterns obtained with PrefixSpan. IGB algorithm checks for each non empty closed[1] pattern, P, if its support is greater or equal to *minconf*. If it is the case, then the generic rule $\emptyset \rightarrow P$ is added to *IGB base*. Else, it iterates on all frequent closed itemsets P_0 subsumed by P. For those having support at least equal to *support(P)/minconf*, the algorithm iterates on the list of minimal generators associated to P_0. During this iteration, we look for the smallest minimal generator g_s, such that there does not exist a generator g_0 subsumed by g_s which is already inserted in the list L of smallest premises. Then, IGB algorithm iterates on all elements of the list L in order to generate rules of *IGB* which have the following form: $g_s \rightarrow (P - g_s)$.

PlanID	Frequent sequential patterns
P1	S1 S2 S4 S5 S6 S7 S8 S9 S10 …
P2	S1 S5 S6 S7 S9 S98
P3	S1 S2 S3 S4 S5 S6 S8 S10 …
P4	S2 S3 S6 S7 S9 S10
P5	S2 S4 S5 S7 S9 S10
P6	S1 S2 S3
P7	S7
P8	S5 S9 S10

Table 3: Part of the meta-context built from data in table 1

Meta-rules	Sup-port	Conf	Expanded meta-rules
S10 ===> S9	4	0.8	…
S9 ===> S7	4	0.8	1 {10 31} ===> 1 {9 11 31}
S9 ===> S5	4	0.8	…
S5 ===> S10	4	0.8	…

Table 4: Examples of generic meta-rules extracted by IGB

By dividing the sub-sequence occurrence by the plans' occurrence, we obtain the relative support associated to the sub-sequence. Let consider a *minsup* of 2 (25%),

[1] Closed subsequences are those containing no super-sequence with the same support.

meaning that a valid sequence occurs in at least 2 input-plans. Table 3 shows the resulting crisp meta-context where dichotomic values (0 or 1) are assigned when a subsequence appears or not in a plan considering the given *minsup*. We simplify by just putting the sequence in the plan entry. Using this meta-context, IGB computes a set of generic meta-rules, part of which is shown in table 4. These meta-rules combined with frequent pattern extend the domain knowledge base that will be used by the tutor.

3. How the Learned Knowledge Base Support Tutoring Services?

Tutoring systems should offer useful tutoring services to assist the learner. These services include coaching, assisting, guiding, helping or tracking the student during problem-solving situations. To offer these services, an ITS need some knowledge related to the context. The knowledge base namely Procedural Task Knowledge (PTK) obtained from the previous framework serves to that end. The next paragraphs present some examples of services that can be supported.

a) Assisting the user to explore possible solutions of a given problem. Domain experts can explore, validate or annotate the PTK. The validation can consist in removing meta-rules with a low confidence, meaning that those rules can not significantly contribute to help the student. Annotation consists in connecting some useful information to meta-rules lattice depicting semantic steps of the problem as well as hints or skills associated to a given step. A meta-rule lattice annotated this way is equivalent to [3]'s Behavioral Network (BN), except that BN is explicitly built from scratch by domain experts. For student users, exploring PTK will help them learn about ways of solving a problem.

b) Tracking the learner actions to recognize the plan s/he is following. Plan recognition is very important in tutoring systems. PTK is a great resource to this process. Each student's action can be tracked by searching the space defined by meta-rules lattice so that we can see the path being followed.

c) Guiding learners. An ITS should be able to help a student solving a problem. A classic help situation is when the student asks what to do next from the actual state. PTK can help to produce the next most probable actions the student should execute and prompt him on that taking into account uncertainty related to rule confidence.

4. Conclusion

In this paper, we proposed a knowledge discovery framework that improves an ITS's knowledge in procedural domain. We used the framework to build a meta-knowledge base of plans from users' traces in CanadarmTutor [4]. The resulting knowledge base enables CanadarmTutor to better help the learner.

References

[1] Pei. J., Han, J. et al (2004). Mining Sequential Patterns by Pattern-Growth: The PrefixSpan Approach. *IEEE Transaction on Knowledge and Data Engineering*, 16(10), oct. 2004.

[2] Gasmi G., Ben Yahia S., Mephu Nguifo E., Slimani Y. (2005). A new informative generic base of association rules. *PAKDD'2005.* LNCS, Springer-Verlag.

[3] Aleven, V., McLaren, B. M., Sewall, J., & Koedinger, K. (2006). "The Cognitive Tutor Authoring Tools (CTAT): Preliminary evaluation of efficiency gains". *ITS'*2006: 61-70. Springer-Verlag.

[4] Nkambou, R., Mephu, E., Couturier, O. and Fournier, P. (2007). Problem-Solving Knowledge Mining from Users' Actions in an Intelligent Tutoring System. Canadian AI'2007. Springer-Verlag.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Analyzing Fine-Grained Skill Models Using Bayesian and Mixed Effects Methods

Zachary A. PARDOS, Mingyu FENG, Neil T. HEFFERNAN, Cristina Linquist-HEFFERNAN
Worcester Polytechnic Institute
{zpardos, mfeng, nth}@wpi.edu

Abstract. Two modelling methods were used to answer the research question of how accurate various grained 1, 5, 39 and 106 skill models are at assessing student knowledge in the ASSISTment online tutoring system and predicting student performance on a state math test. One method is mixed-effects statistical modelling. The other uses a Bayesian networks machine learning approach. We compare the prediction results to identify benefits and drawbacks of either method and to find out if the two results agree. We report that both methods showed compelling similarity which support the use of fine grained skill models.

1. Introduction

Intelligent Tutoring Systems (ITS) rely on models that associate the various skills students are learning with different questions or actions. We have created 3 different models at different grain sizes. One model has 5 skills, another 39 and our finest grain sized model has 106 skills. A model with a single skill is also used to represent unidimentional assessment. We have found that working with the 20 teachers that use our system, many appreciate the reports made possible by the fine grain sized models that tell them which specific skills a student is doing poorly on. But are these finer grained models at least as accurate as more traditional, coarse grained models [1]?

This paper attempts to compare two ways of modelling skills; *Bayesian networks* [2], popular in Artificial Intelligence research, and *mixed-effects modelling* [3], popular in statistical departments. We investigate both modelling methodologies's result to the question of "Are finer grain sized skill models more accurate at test prediction". We will be able to gain confidence in our approach if both methods corroborate each other's result. We will use a dataset of online tutoring responses from 447 students and their respective state test scores to conduct the evaluation.

1.1. Background on the MCAS State Test and ASSISTment Project

We will be evaluating our models by using scores from the 8th grade 2005 Massachusetts Comprehensive Assessment System (MCAS) mathematics test which was taken after the online tutoring data was collected. The ASSISTment system is an e-learning and e-assessing system [4]. In the 2004-2005 school year 8 classes of students used the system about once every two weeks. Students were presented with randomly selected 8th grade math items to answer. Each tutoring item, which we call an ASSISTment, is based upon a state released MCAS item (from previous state tests) which we have added "tutoring" to. We believe that the ASSISTment system has a better chance of showing the utility of fine-grained skill modelling due to the fact that we can ask scaffolding questions that break the problem down in to parts and allow us to identify what skill the student is having trouble with. As a matter of logging, the student is only marked as getting the item correct if they answer the question correctly on the first attempt without assistance from the system.

1.2. Creation of the Fine-Grained Skill Model

A seven hour "coding session" was staged at Worcester Polytechnic Institute (WPI) where our subject-matter expert, Cristina Heffernan, with the assistance of the 3rd author, set out to make up skills and tag them to all of the 300 existing 8th grade MCAS items from previous years. Because we wanted to be able to track learning between items, we wanted to come up with a number of skills that were somewhat fine-grained but not too fine-grained such that each item had a different skill. We imposed upon our subject-matter expert that no one item would be tagged with more than 3 skills. She gave the skills names, but the real essence of a skill was what items it was tagged to. This model is referred to as the 'April' model or the WPI-106. The National Council of Teachers of Mathematics and the Massachusetts Department of Education use broad classifications of 5 and 39 skill sets. The 39 and 5 skill classifications were not tagged to the questions. Instead, the skills in the coarse-grained models were mapped to the finer-grained models in a "is a part of" type of hierarchy, as

opposed to a prerequisite hierarchy [5]. The appropriate question-skill tagging for the WPI-5 and WPI-39 models could therefore be derived from this hierarchy.

2. Methodologies

2.1. Bayesian Method

The Bayes nets consist of 3 layers of binomial random variable nodes. The top layer nodes represent skill knowledge with a background probability set to 0.50, while the bottom layer nodes are the actual question nodes with conditional probabilities set to 0.10 for guess and 0.05 for slip. The intermediary 2nd layer consists of ALL gates that, in part, allow us to only specify a guess and slip parameter for the question nodes regardless of how many skills are tagged to it. The guess and slip parameters were not learned but instead set ad-hoc. When we later try to predict MCAS questions, a guess value of 0.25 will be used to reflect the fact that the MCAS items being predicted are all multiple choice, while the online ASSISTment items have mostly been converted to text-input fields. Future research will explore learning the parameters from data.

A prediction evaluation is run for each model one student at a time. The student's responses are presented to the Bayes net as evidence and inference (exact join-tree) is made on the skills to attain knowledge probabilities. To predict each of the 39 questions on the test we used the inferred skill probabilities to ask the Bayes Net the probability of the student getting the question correct. We get a predicted score by taking the sum of the probabilities for all questions. Finally, we find the percent error by taking the absolute value of the difference between predicted and actual score and dividing by the total possible points on the test (39). The average error of a skill model is the average error across the 447 students.

2.2. Mixed-Effects Method

For dichotomous (binary in our case) response data, several approaches adopting either a logistic or probit regression model and various methods for incorporating and estimating the influence of the random effects have been developed. As indicated in [6, 7], the mixed-effects logistic regression model is a very popular and widely accepted choice for analysis of dichotomous data. It describes the relationship between a binary or dichotomous outcome and a set of explanatory variables. Such a model is often referred to as “longitudinal model” [3] since time is introduced as a predictor of the response variable, which allows us to investigate change over time. After the model was constructed, the fixed-effects for the whole group and the random effects for each student were extracted and then the two learning parameters “intercept” and “slope” were calculated for each individual student (and for each skill if skill was introduced as a factor into the model). Given this, we can apply the model on the items in the state test to estimate students’ response to each of them.

3. Results

Both methods used the same data and the same skill taggings [4, 8] to evaluate users. The MAD score is the mean absolute difference between the predicted and actual score. The under/over prediction is our predicted average score minus the actual average score on the test. The actual average score will be the same for all models. The centring is a result of offsetting every user’s predicted score by the average under/over prediction amount for that model and recalculating MAD and error percentage.

Table 1. *Bayesian and Mixed Effects Test Prediction Results*

Model		Error	MAD Score	Under/Over Prediction	Error (after centering)	Centered MAD Score
WPI-106	Bayes	13.75%	4.19	↓ 1.0	13.60%	4.10
	Mixed-effects	12.10%	4.12	↓ 0.6	12.04%	4.10
WPI-39	Bayes	12.05%	4.10	↓ 1.0	11.82%	4.02
	Mixed-effects	12.40%	4.22	↑ 1.2	12.07%	4.11
WPI-5	Bayes	18.70%	5.42	↓ 3.5	16.20%	4.70
	Mixed-effects	12.84%	4.37	↑ 1.8	12.13%	4.12
WPI-1	Bayes	25.46%	7.39	↓ 4.6	21.63%	6.27
	Mixed-effects	13.00%	4.42	↑ 2.1	12.06%	4.10

We can see that the mixed-effects models outperformed all but the Bayesian 39 which stands as the highest performing model overall. A sizeable decrease in accuracy can be observed from the mixed-effects 5 and 1 to the Bayesian 5 and 1. This result suggests that the current Bayesian prediction performs best with the finer grained models. The top performing models for both methods are the fine-grained WPI-39 and WPI-106. The relative performance of each grain sized model is the same for each method with the exception of the 39, which is the Bayes best performer, and 106, which is the mixed-effects best performer. However, the results from these two models are not statistically different. A paired T-test between the 39 and 106 of each method gives a value of 0.828 for Bayesian and 0.215 for mixed-effects. All other models compared to one another are statistically significantly different with P values inferior to 0.05. Residual plots of Bayes and mixed-effects models were analyzed and proved to be very similar. These similarities raise our confidence in the execution of the methodologies.

4. Conclusions

We have seen how the methods of Bayesian networks and mixed-effects modelling produce very similar results. This gives us confidence in the execution of the two methods and provides a rare doubly reinforced result arrived at from two different angles. We have also answered our research question about the utility of finer-grained models and can report that the fine-grained WPI-39 and WPI-106 models provide the most accurate test prediction as confirmed by both Bayesian and mixed-effects methods. Teachers will be happy to know that there is some validity to the fine-grained models.

There is one implication we would like to discuss. In the United States many schools and teachers are being encouraged to use frequent (i.e., monthly) testing to be "data-driven". The problem is that many tests used are unidimensional and do not provide cognitive reporting to the teachers while at the same time taking up valuable class time. There seems to be a tension between tests that are fast and tests that are cognitively diagnostic. One implication of the success of fine grained model assessment is that it might be possible for states to use these models to develop a system of their own similar to ASSISTment that does all three of these things; 1. accurately assesses students, 2. gives finer grained feedback that tends to be more cognitively diagnostic and 3. saves time by assessing students while they are getting "tutoring" or taking a test.

Acknowledgements

We would like to thank all the people associated with creating the Assistment system listed at www.assistment.org including the investigators Kenneth Koedinger and Brian Junker at Carnegie Mellon. We would also like to acknowledge funding from the US Department of Education, the National Science Foundation, the Office of Naval Research and the Spencer Foundation. All of the opinions expressed in this paper are those solely of the authors and not those of our funders.

References

[1] Hambleton, R.K., & W. J. van der Linden. (1997). Handbook of modern item response theory. New York, NY: Springer-Verlag.

[2] Reye, J. (2004). Student modelling based on belief networks. International Journal of Artificial Intelligence in Education: Vol. 14, 63-96.

[3] Singer, J. D. & Willett, J. B. (2003). Applied Longitudinal Data Analysis: Modeling Change and Occurrence. Oxford University Press, New York.

[4] Feng, M., Heffernan, N. T., Mani, M., & Heffernan, C. (2006). Using Mixed-Effects Modeling to Compare Different Grain-Sized Skill Models. In Beck, J., Aimeur, E., & Barnes, T. (Eds). Educational Data Mining: Papers from the AAAI Workshop. Menlo Park, CA: AAAI Press. pp. 57-66. Technical Report WS-06-05. ISBN 978-1-57735-287-7.

[5] Carmona1, C., Millán, E., Pérez-de-la-Cruz, J.L., Trella1, M. & Conejo, R. (2005) Introducing Prerequisite Relations in a Multi-layered Bayesian Student Model. In Ardissono, Brna & Mitroivc (Eds) User Modeling 2005; 10th International Conference. Springer. 347-356

[6] Snijders, Tom A. B., and Bosker, Roel J. (1999). Multilevel Analysis: An Introduction to Basic and Advanced Multilevel Modeling, London etc.: Sage Publishers, 1999.

[7] Hedeker, D. & Gibbons, Robert. D. (in progress). "Longitudinal Data Analysis": "Mixed-Effects Regression Models for Binary Outcomes" (chapter 9).

[8] Pardos, Z. A., Heffernan, N. T., & Anderson, B., Heffernan, C. L. (accepted). The Effect of Model Granularity On Student Performance Prediction Using Bayesian Networks. User Modelling 2007: 11th International Conference.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Automated Assessment of Problem Difficulty in Mathematical Analysis

Ildikó PELCZER[1] and Fernando GAMBOA
UNAM, Circuito Exterior s/n, 04510, Coyoacan, Mexico D.F., Mexico

Abstract. It is common between teachers of mathematics to assess the difficulty of the problems they gave to students. The evaluation is mostly based on subjective appreciation with no explicit consideration of difficulty factors. However, in a computerized environment for mathematics instruction it would be highly beneficial to dispose of such explicit factors and automated assessment of problem difficulty. In the present paper a set of factors for sequence problems in analysis are proposed. For the automated detection of these factors a set of rules was defined. Knowledge representation and implementation details are described.

Keywords. Problem difficulty, mathematical analysis, automated assessment.

Introduction

It is common between teachers of mathematics to evaluate the difficulty of problems chosen for students. Traditionally, problem difficulty is measured by *item difficulty ratio* (the ratio between the number of students who resolve correctly a problem and the total number of students, [1]). Another common measure is *response time* that is time passed between the presentation and resolution of an item [2]. These measures are domain-independent; however, their application requires previous studies.

In this paper we propose a set of cognitive difficulty and complexity factors related to sequence problems in mathematical analysis. The investigation is part of a larger project that has the aim of defining a computer model of human problem generation in mathematics. In that context, difficulty assessment plays an important role, since teachers often consider it as guiding line for the generation. The implementation of an automated evaluation asks for a proper knowledge representation of a problem and for rules that describe problem contexts in which those factors can be identified.

1. Difficulty Factors in Mathematical Analysis

There is a large body of research on difficulties encountered by students while learning analysis (see, for example, [3, 4, and 5]). Based on these studies we propose six cognitive and three complexity factors for sequence problems. Cognitive factors relate to general mathematical abilities or to the way in which concepts are acknowledged and internally kept. On other hand, complexity factors relate more to algebraic

[1] Corresponding author: Ildikó Pelczer, UNAM, Circuito Exterior s/n, 04510 Coyoacan, Mexico D.F., Mexico; E-mail: IPelczer@iingen.unam.mx

procedural knowledge or specific rules that can be applied for a domain. First, the six cognitive difficulty factors are described, followed by three complexity factors.

Dual The factor is defined in order to underline difficulties arising from the dual nature of mathematical concepts (object-process). In sequence problems is present whenever there is a need to perform operations on the limit.

Metric The metric factor is proposed here to capture the difficulty in handling other spaces than the real or other metric than Euclidian one.

Logic The factor is proposed in order to capture difficulties that students have in the maneuver of quantifiers and logical propositions. It is considered present in a sequence problem whenever the solution of the problems implies their use.

Parameter The factor denotes the presence of a parameter in a problem. Such problems often require from the student to identify interesting sets for parameter values. When specified, the set of parameter values is, frequently, subset of rationales.

Form This factor stands for the difficulty in handling sequences in which the general term is given by several expressions, recurrence relations or by a property.

Reasoning The factor illustrates the difficulties related to inductive or proof by contradiction types of reasoning.

Number of concepts It refers to the number of concepts involved in the expression.

Number of required partial results The factor denotes the partial results (for the application of domain specific rules) to be obtained in order to solve the problem.

Number of transformations The factor stands for the transformations to be applied on problem expression in order to reduce it to something already known.

2. Knowledge Representation, Rules and Implementation

The implementation requires the identification of problem contexts in which specific factors are present and representation of domain general and specific knowledge.

In the system a problem is described by its type, given and requested elements. *Type* takes value from a set of predefined values and has an associated text used in the textual formulation to the problem. *Given* part contains two further blocks: *Sequence* (description of the sequence) and *Properties* (given relations). The *Requested* part contains two blocks: *Property* and *Restriction.* The first one contains the entity to be determined; the second, restrictions that refer to the first one (table 1, section 1).

The description contains keywords (in majuscule in the example) that are stored separately in the knowledge base and have a brief textual description attached to it. The text is used to replace the keyword when it comes to textual formulation of the problem.

For an automated assessment we have to describe and formalize the problem contexts in which those factors were identified. There were analyzed 460 sequence problems and annotated in each the identified factors. Problems were then regrouped based on detected factors. The rules describing the situations in which a factor is present were established by analyzing each group. In section 2 two examples of context-factor description are presented. Similar descriptions exist for all the defined factors.

Rules for factor presence detection are based on the correspondence described above, however, further conditions are needed such to describe in more detail each context. For example, the rule that establishes if the *partial result* factor is present due to the application of Cauchy criteria has to contain criteria that specifies formally when a sum is not computable. The conditions that describe when a sum is not computable

are stored separately from rule description. They are formulated as a set of conjunctive criteria (section 3). We give as example the rule for application of the Cauchy's criteria. A sum is considered as not computable if the general term can't be simplified, nor can be simplified the difference of two consecutive terms, nor can be simplified after a transformation (usually logarithmic or trigonometric) of the expression under the sum.

Table 1. Problem representation

Section 1: Problem description example	**Description of elements**
PB_1	COMPUTE : PbType
PbType COMPUTE	Text
Given	REPEAT Req_x : "Compute" Req_x.Prop &
Seq : Index n	KEYWORD.Text & Req_x.Restr
Type NATURAL	Etext
Term u(n) v(n)	*Final text*:
Range REAL REAL	Given two sequences u(n), v(n) such that v(n)=Ln(u(n)-
Expr.Form NULL LN(SEQ(u_n)-1)	1), compute the general term of u(n) such that u(1)=2 and
ESeq	v(n) is an arithmetical sequence.
EndGiven	
Req	
Prop GENTERM u(n) EProp	
Restr GIVENTERM u_1=2	
ARITMSEQ v(n) ERestr	
EndReq	
END_PB_1	
Section 2: Factor	**Element in the problem description**
Cognitive factor: Metric	
- explicitly stated limit request for non-numerical sequence	-Seq.Range AND Req.Prop
Complexity factor: Number of concepts	
Concepts involved in the expression	- Expr.Form
Section 3: Rule	**Description of not computable sum**
FCompl2= Criteria Cauchy	Not_Comp (SUM)
Condition	Not(Simplify(SUM.Expr(Index)))
Seq.Expr.Form=SUM &&	Not(Simplify(SUM.Expr(Index)-SUM.Expr(Index-1)))
Not_Comp(SUM)	Not(Simplify(Transform(SUM.Expr(Index)))

In the implementation a set of text files (containing keywords, rules, context descriptions, etc.) communicate with an user-interface that, on its turn, sends expressions to specialized components (like MatLab, MATHEMATICA) to decide their value or characteristics. Once the factors of a problem are identified they are stored in a text file that contains the problem, factors present and the rules applied during the assessment process.

References

[1] N. E. Gronlund. Measurement and evaluation in teaching. Macmillan, New York, 1981.

[2] E. Mason et al. Response time and item difficulty in a computer-based high school mathematics course. Focus on LearningProblems in Mathematics, 14(3), pp. 41-51, 1992.

[3] B. Cornu. Apprentissage de la notion de limite. Modèles spontanés et modèles propres. In 5th Annual Meeting of the International Group of Psychology of Mathematics Education, 1981.

[4] A. Sierpinska. Humanities students and epistemological obstacles related to limits. Educational Studies in Mathematics, 18, pp. 371-397, 1987.

[5] D. Tall and L. Schwarzenberger. Conflicts in the learning of real numbers and limits. Mathematics Teaching, 82, pp. 44-49, 1978.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Automatically Generated Inspectable Learning Models for Students*

Diana PEREZ-MARIN, Ismael PASCUAL-NIETO, Enrique ALFONSECA, Pilar RODRIGUEZ
Computer Science Department, Universidad Autonoma de Madrid

Abstract. Willow is an automatic and adaptive free-text scoring system. Its main purpose is to provide formative assessment to the students so that they can get more practise and feedback before their exams. It also keeps track of the terms that students use in their answers to automatically generate each particular student conceptual model and the group conceptual model. In both cases, the conceptual model can be defined as a set of concepts and their relationships that each student keeps in his or her mind about an area of knowledge. The conceptual model can be represented by COMOV as a concept map so that instructors can track students' concepts acquisition. In this paper, we present the new possibility and its consequences of showing the concept maps (from their individual and group conceptual models) not only to instructors but also to students. Moreover, to allow students to see their evolution, as they study and get more training, through their concept maps. The preliminary results of an experiment carried out with a group of students show how they like to be able to inspect their generated learning models.

Keywords. Inspectable students' conceptual models, e-assesment, e-learning

Introduction

Concept maps are powerful tools to visually represent the conceptual structure that someone has about an area of knowledge [1]. They have been used as an aid for people to conceptualize and organize their knowledge. However, they are not being extensively applied in classrooms nowadays. It could be due to the fact that to learn how to create them is time consuming and it is difficult to manage them in paper.

An attempt to solve these problems has been to support the management of concept maps with educational computer systems such as VisMod [2] and DynMap[3].

Our approach releases the students of the task of having to introduce the map into the computer, as the concept map is automatically generated from answers to questions by an automatic free-text scoring system and shown to the instructors with COMOV, an on-line system in which they can log to see the current state of a particular student or the whole class. In this paper, we present an improvement of the procedure, inspired in the work presented in [4], so that not only teachers can see the generated concept maps but also students. It allows to achieve a higher quality of diagnosis as the student is more involved, to get additional feedback and to promote reflective thinking by allowing students to ask the teacher about their generated models.

* This work has been sponsored by Spanish Ministry of Science and Technology, project TIN2004-03140

1. Automatically generated inspectable learning models

Willow [5] is a web-application[1] able to assess free-text answers written both in Spanish and English. It has an associated authoring tool to allow instructors to introduce questions and a set of several correct answers per question. Questions are grouped into topics and topics into collections belonging to certain areas of knowledge.

Terms are extracted from references using an automatic procedure [5]. A term can be considered as the label of a concept, and it is usually defined as a word or a multi-word expression used in specific domains with a specific meaning. We have identified three different types of concepts: basic concepts (BCs) that correspond to the terms found in the references; topic concepts (TCs) that are the name of the lessons; and, area-of-knowledge concepts (ACs) that are the name of the courses.

Whenever a student answers a question in Willow, the system keeps track of the use of the identified terms and assigns them a confidence value (CV) of how well the student knows this concept according to the heuristics explained in [5].

BCs can be related to other BCs or to TCs. Moreover, TCs can also be related to one AC. TCs have also associated a CV that is calculated as the mean of the CV of the BCs related to it. The same is applicable to ACs. Willow stores this information so that when instructors or students log into COMOV, they can see the students' conceptual models represented as concept maps in which the nodes are BCs, TCs and ACs and the links relate BCs-BCs, BCs-TCs and TCs-ACs. A size and color scheme has been used to give information about the type of concept represented and its confidence-value. According to the size, bigger nodes are ACs, medium-size nodes are TCs and small nodes are BCs. Regarding the color, it goes from green (absolute confidence about that the concept is known) to red (absolute confidence about that the concept is unknown).

Traditionally, only teachers were able to inspect this concept map. Inspired by the work of Dimitrova [4], we have also given access to COMOV to the students so that they can see their individual and group conceptual models and organize their study. That is, the concepts that appear in green are known and thus, students should not review them again, whereas concepts in yellow or red need to be reinforced.

2. Experiment and results

In the first semester (from October 2006 to January 2007), we carried out an experiment with Willow and COMOV with a group of 24 volunteer students of the subject "Operating System" of the fourth academic course of the Telecommunications Engineering grade. To motivate the students to use Willow, they were told that they could practise with questions from exams of past years with the references of the instructors, they could also see their conceptual model as an automatic generated concept map and it will positively count for their final score.

First of all, we did an overview questionnaire with two open items: a) I would like to see my generated concept map because… and b) I would not like to see my generated concept map because…

Eight students answered the questionnaire. All of them said that they would like to see their generated concept maps because it would allow them to track their knowledge acquisition about the subject.

[1] Available at http://orestes.ii.uam.es:8080/ateneaAdaptativa/jsp/loginAtenea.jsp

Our insight was also that the concept map would help. Thus, after they had finished one or more topics with Willow, we allowed them to inspect their generated concept maps and tell us what they think about them. Students answered that they find concept maps very useful because it clearly showed their conceptual structure. They also like to see the evolution of the concept maps as they progress in their study.

For instance, see in Figure 1 the evolution of the generated model of a student from the 10th of October 2006 in which she only knew about introduction and a little about processes to the concept map generated the 1st of December 2006 in which she has learnt more about processes and threads. It can be observed how in the right figure, there are more nodes with lighter colour indicating that the system is more confident that the student knows the concepts of the course. Moreover, it can also be observed how the structure of the concept map has changed. It is because new links between concepts have been identified from the new answers introduced in the system by the student.

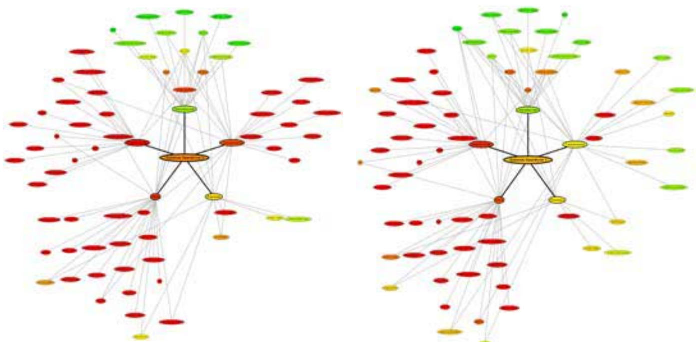


Figure 1. Example of the evolution of the generated concept maps of a student

3. Conclusions and future work

Showing their generated concept maps to students not only helps them in organizing their study but also motivate them to continue learning as they can check how their concept map evolves. In the future, we plan to carry out a more detailed experiment with the students in which not only the concept map representation of the conceptual model is studied but also other implemented representations such as conceptual diagrams of knowledge, tables, graphs, or textual summaries.

References

[1] Novak, J. & Gowin, D. (1984), Learning How to Learn, Cambridge Univ. Press, U.K.

[2] Zapata-Rivera, J. (2004), 'Interacting with Inspectable Bayesian Student Models', International Journal of Artificial Intelligence in Education 14(2), 127-163.

[3] Rueda, U.; Larranaga, M.; Elorriaga, J. & Arruarte, A. (2004), 'Validating DynMap as a mechanism to visualize the student's evolution through the learning process', LNCS, 864-866.

[4] Dimitrova, V. (2003), 'STyLE-OLM: Interactive Open Learner Modelling', International Journal of Artificial Intelligence in Education 13(1), 35-78.

[5] Perez-Marin, D.; Alfonseca, E. & Rodriguez, P. (2006), Automatic Generation of Students' Conceptual Models underpinned by Free-Text Adaptive Computer Assisted Assessment , in 'Proceedings of the IEEE International Conference on Advanced Learning Techniques (ICALT)'.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

In-game help in the Tactical Language Training System

Prasan SAMTANI, W. Lewis JOHNSON

USC Information Sciences Institute
4676 Admiralty Way, Marina Del Rey CA 90292
{samtani, johnson}@isi.edu

Abstract

Game-based learning software shows tremendous potential in its ability to motivate and engage learners. However, when giving pedagogical feedback in a game context, it is important to do so in a way that does not break a student's engagement and is not prone to abuse. This paper describes in-game help in the Tactical Language and Culture Training System, a tool designed to help learners rapidly acquire basic communicative skills in a foreign language and culture.

1 Introduction

The Tactical Language and Culture Training System (TLTS) helps learners acquire basic communicative skills in a foreign language and culture. The system is broadly divided into two main sections. In the Skill Builder, learners are coached through a set of lessons on language and culture by a virtual tutor. The tutor offers attributional and motivational feedback in order to keep the learner on track[3], and maintains a model of the learner's current skill set.

Following the acquisition of skills, learners must proceed to complete missions in a simulated environment populated with virtual characters - the Mission Game. Learners can speak to AI characters in the game through a microphone, and can select appropriate gestures using the mouse. In order to successfully complete missions, the learner must display a mastery of the specific linguistic and skills in the target language, as well as knowledge of the culture. The learner is accompanied by an aide character, who can offer recommendations if the player gets stuck.

Two training systems have been built so far using TLTS: Tactical Iraqi (for Iraqi Arabic) and Tactical Pashto (for Pashto, spoken in Afghanistan). Additional courses for other languages are under development. Tactical Iraqi is currently in use by thousands of learners in the US armed forces as they prepare for deployment overseas. Reports from users suggest that the program is engaging and highly effective. Anecdotal reports indicate that trainees start acquiring basic communication skills very rapidly, and acquire basic functional proficiency in conversational Arabic relevant to their mission in just a few weeks. However, there were some reports that learners sometimes made inappropriate use of the courseware by misusing the hint system. The remainder of this paper describes the new help system that was designed to counteract this misuse.

Figure 1: The Tactical Language Mission Practice Environment: The aide character is to the far left

2 Description

A major problem with the provision of in-game assistance is that users tend to use it unproductively[1, 2]. In the initial version of the system, learners could ask for help by requesting the character for a hint. The system would examine the state of the multiagent system and provide the best action under the circumstances. Although the overall system was found to be highly effective, learners could abuse the help system by consistently asking for hints. Additionally, the earlier system made users feel as if there was only a single correct path through a scenario, and we felt it did not do enough to encourage exploration. The new help system was designed specifically to order to address these shortcomings.

Our criteria for designing the new system were that (1) the system should encourage the user to explore the scenario and use a variety of phrases, (2) hint abuse should be detected and dealt with organically, and (3) that the user should be able to understand what went wrong after going through the scene.

Figure 2 details the architecture of the in-game help system in TLTS. When the user asks for help, the request is sent to a hint handler, which studies the state of the multiagent system and the previous user actions and provides a hint. Any action that can move the story forward is put into the pool of possible actions. The hint handler then selects a single action from the pool to recommend - actions that the user has already succesfully executed are less likely to be picked, this encourages the users to try out phrases that they have not yet mastered.

The hint object is then passed on to an abuse detector, which examines the users help-seeking history and skill model to look for signs of abuse. Abuse is generally detected if the user consistently asks for help or if the skill model shows that the user has the capability to come up with a response without help (based on previous competence displayed in the Skill Builder). If abuse is mild, for example, if the skill model shows that the user has mastered a skill but is still requesting assistance, the hint is given in English instead of the target language. If abuse is extreme, the multiagent system is informed, and the ability to succeed in the scene is negatively impacted. As an example of extreme abuse,

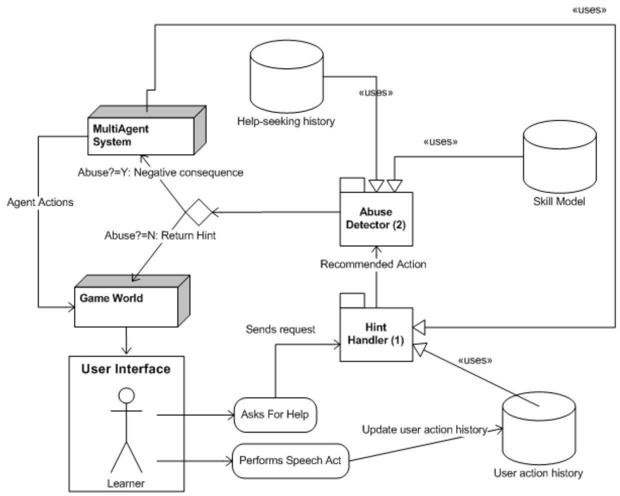


Figure 2: The in-game help system

if the user repeatedly asks the aide character for a hint enough times within a scenario (and the skill model shows that the user has completed the relevant portions of the Skill Builder), the NPCs begin conversing directly with the aide. Upon debriefing, the behavior of the NPCs is explained to the user.

3 Future Work

The system described above has been implemented in the TLTS. We plan to conduct studies on the effectiveness and user-friendliness of the system. Users will be provided with the improved version of the system, and the data will be examined and compared to the data collected with the earlier system. We are particularly interested in examining whether the system (1) encourages users to try new paths through the scenarios and (2) whether the abuse detection reduces users' reliance on hints. Another area of future work is to more closely integrate the in-game help with the multiagent system.

References

[1] Aleven, V., McLaren, B., Roll, I., and Koedinger, K. Toward tutoring help seeking. *In proceedings of the Seventh International Conference on Intelligent Tutoring Systems* (2004), 227–239.

[2] Aleven, V.and Koedinger, K. Limitations of student control: Do students know when they need help? *In proceedings of the 5th International Conference on Intelligent Tutoring Systems* (2000), 292–303.

[3] Johnson, W.L., Beal, C., Fowles-Winkler, A., Lauper, U., Marsella, S., Narayanan, S., Papachristou, D., Valente, A., and Vilhjalmsson, H. Tactical language training system: An interim report. *In proceedings of Intelligent Tutoring Systems* (2004).

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Supporting Learning Design via dynamic generation of learning routes in ADAPTAPlan

Olga C. SANTOS [a,1], Jesus G. BOTICARIO [a]
[a] *aDeNu Research Group, Artificial Intelligence Dept., CSS, UNED*

Abstract. The design phase on the life cycle of the eLearning process is currently one of the main bottle necks in adaptive systems. To facilitate this process and reduce the design effort our approach focuses on providing dynamic assistance to some of the author's tasks that strongly depend on data coming from users' interactions. In particular, ADAPTAPlan project is developing a system where the learning route of a student is dynamically built from the combination of user modelling and scheduling techniques making a pervasive use of educational standards (IMS). The author is requested to provide simple information about the course structure, pedagogy and restrictions and the system utilize this information together with the user model to generate a personalize IMS-LD course flow suited to that learner. Since the output is given in a standardized format, the course can be run in any standard based learning environment. This approach will be tested on a real course, "How to teach online", within the UNED's program for ongoing education and open courses.

Keywords. Learning Design, User Modelling, Metadata and Learning, Learning Objects, Learning Activities, Educational standards, Design templates, Adaptive eLearning

1. Introduction

Nowadays it is generally accepted that learning should be a personalised process. In this line, some research projects, such as OPAL, OLO and KOD [1] intend to extend existing educational standards to support adaptive course delivery while others, such as EU4ALL, ALPE are addressing students' individual needs to support eInclusion. All these projects address a critical design issue, which is to determine a priori all the possible situations that may arise during the course execution, including learning materials, pedagogical models, learning styles and learning needs. However, not everything can be specified in advance by the author because unexpected situations appear at run time that cannot be predicted at design time. Furthermore, even knowing everything in advance does not suffice because of the management problems involved, i.e., describing all the existing possibilities and making the adaptation process sustainable over time. In order to cope with these problems, design-time and run-time approaches were combined in aLFanet project in terms of an extensive use of educational standards (IMS-CP, IMS-LD, IMS-QTI, IMS-MD, IMS-LIP). In this approach, the design created in IMS-LD is central in the learning process and the evaluation performed showed that course authors experienced the design phase as a complex task [2].

[1] aDeNu Research Group, Artificial Intelligence Dept., Computer Science School, UNED, C/Juan del Rosal, 16, 28040 Madrid, Spain. E-Mail: {ocsantos, jgb}@dia.uned.es –http://adenu.ia.uned.es

A proposal to tackle this problem came from our experience in a previous project on e-tourism. In SAMAP project (TIC2002-04146-C05) planning and user modelling techniques were combined to provide adapted routes to visit a city. Based on this proposal, ADAPTAPlan project (TIN2005-08945-C06-01) was funded to analyse the capability of automatically generate learning routes adapted to the students' needs by integrating scheduling, machine learning and user modelling techniques.

2. ADAPTAPlan approach

To reduce the design effort of the author, we are researching the way to provide dynamic assistance to support this process and ask the author to focus on those elements that require the author's experience and expertise. This approach differs from other related course generation approaches based on planning [3, 4] or late modelling [5] and asks the author to focus on the learning process in terms of objectives and learning activities, with their corresponding learning objects properly characterized in IMS-MD. They also consider the educational services (i.e., forums, calendars, document storage spaces, etc.) that support the activities and a set of conditions, initial requirements and restrictions in IMS-LD level B. From the student point of view, the system is going to take into account the individual initial knowledge, motivations, learning styles and learning goals, as well as the interaction data gathered and inferred from the students' behaviour in the course. This information is also specified in IMS-LIP. The user model is based on explicit information provided by the student through questionnaires (in IMS-QTI) and/or data learnt from the analysis of the student's previous interactions with machine learning techniques. With the information known about the student and the description of the learning tasks defined by the author, the system can schedule the tasks and offer the student a learning plan (equivalent to the concept of learning route from instructional design) adapted to the student's needs. In fact, this plan is specified in IMS-LD, so it can be run in a standard-based learning management system. The individual student and the student group behaviour are to be monitored to feed back the user model and scheduling systems for next runs. Moreover, the author of the course is provided with reports on the students' performance, which can affect the tasks design, closing the life cycle of adaptation in eLearning as defined in [2]. This approach goes further than others that consider providing the output in a similar structure to IMS-CP [3].

More detailed, ADAPTAPlan approach can be summarized in the following steps:

1. The author defines initial requirements such as tasks, milestones, restrictions, services and characterizes the course materials.
2. The planner takes these requirements together with the user model of a particular learner and defines a linear plan which consists on a particularized learning route (in terms of IMS-LD) generated for that learner.
3. This learning route is loaded into the course for the learner. Activities, learning objects and services are offered to the learner according to the restrictions specified in the learning design.
4. This IMS-LD is extended to include in the specification all the resources (activities, services, learning objects) available in the course, and not only those included in that particular learning design. This is required in case the

learning route has to be re-planned because the initial planning of the course fails or gets to a stop point (see 5).

5. If the plan reaches an impasse due to: (i) stop point defined by the author (e.g. an evaluation or a temporal restriction), (ii) fails because the learner diverts from the planned route, (iii) blocks in a point of the course, or (iv) the learner cannot perform an activity, the planner modifies the initial plan taking into account the runtime information.
6. From that moment on, the planner (taking into account the full structure loaded in point 4) guides the learner in the course based on the runtime information.
7. When the course ends, the interactions of all the learners are analyzed and used to build a generic IMS-LD. This learning design abstracts all the particularized initial IMS-LD together with the runtime modifications and builds an IMS-LD with all the possible learning routes in terms of properties.

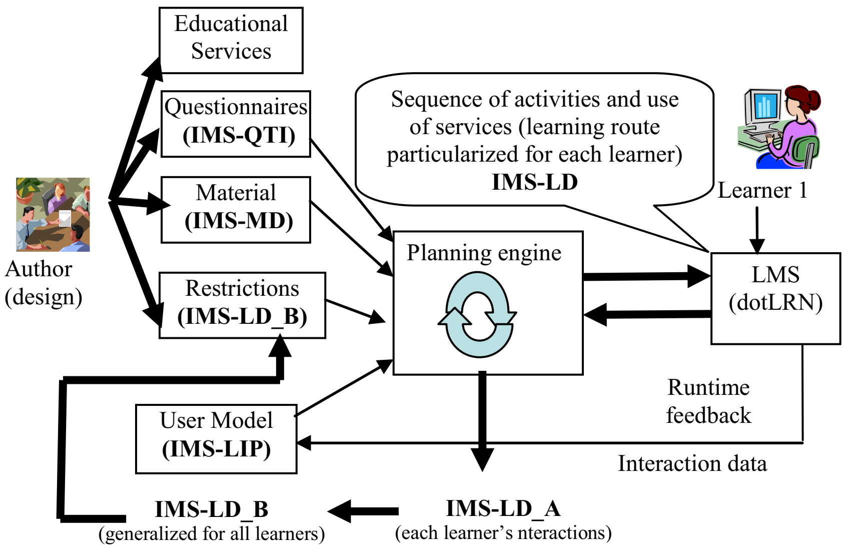


Figure 1. General overview of the ADAPTAPlan approach

References

[1] Paramythis A., Loidl-Reisinger S., and Kepler J. Adaptive Learning Environments and e-Learning Standards. Electronic Journal of eLearning, EJEL: Vol 2. Issue 1, March, 2004.

[2] Boticario, J.G., Santos, O.C.. Issues in developing adaptive learning management systems for higher education institutions. International Workshop on Adaptive Learning and Learning Design, Adaptive Hypermedia 2006. Ireland, 2006.

[3] Brusilovsky, P. Vassileva, J. Course sequencing techniques for large-scale webbased education. Int. J. Cont. Engineering Education and Lifelong Learning, Vol. 13, Nos.1/2, 2003.

[4] C. Ulrich. Course generation based on HTN planning. Proc. of the 13th Workshop of the SIG Adaptivity and User Modeling in Interactive Systems, pp. 74-79, 2005

[5] Zarraonandia, T. Fernández, C., Dodero, J.M. A late modelling approach for the definition of computer-supported learning processes. International Workshop on Adaptive Learning and Learning Design, Adaptive Hypermedia 2006. Ireland, 2006.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Developing Seamless Learning Environment for Primary Science Education

Peter SEOW, Chee-Kit LOOI, Baohui ZHANG
National Institute of Education, Singapore

Introduction

Based on preliminary results from an on-going research project at the Learning Sciences Lab in Singapore (Zhang et al., 2006), we propose an initial design framework to integrate student inquiry and discovery activities, and the use of mobile and web-based technologies with intelligent support for learning. The intention for the research is to design a seamless learning environment which is theoretically aligned with the lenses of Distributed Cognition, and of Community of Learners. The following questions guide our design and research work: How to design inquiry-based learning activities that make use of mobile and web-based learning technologies for primary science students? How to design a seamless learning environment that integrates the above mentioned technologies?

1. Mobile Learning in Environmental Science Education

Environmental education becomes very important for citizens to make decisions in their daily life; efforts are being made by countries world-wide to increase the awareness of these issues among school children. In Aug 2006, we worked with 6 primary schools in Singapore to implement a technology-enhanced learning (TEL) environment to help 480 primary four students to learn about the 3Rs (Reduce, Reuse, and Recycle). The objective of the project was to use mobile and web-based technologies to enable the children to learn the concepts of 3Rs through a series of activities held over time in different locations. The series of activities culminated with the students sharing with others on their plans or efforts in conserving the environment through the 3Rs. A customized Pocket PC application integrated with a built-in camera was designed and developed to allow students to go through iterative learning cycles. The design intended to engage students cognitively by recording their prior knowledge, observations, interaction with the public, their plans on improving the environment and application of their plans. Based on data from one of the six schools, the results of the project seemed to be encouraging. We recorded significant gains in the pupils' understanding of the 3Rs through pre-test and post-tests. Interviews with a target group of pupils and student groups' classroom presentations revealed that they were more aware of the environment problems and spoke to their family about being more environmentally friendly by actually taking some measures to promote 3Rs, such as using reusable bags when the family shops at the supermarket (Zhang et al., 2006).

2. Design Framework

With the anticipation of personal mobile devices and wireless network infrastructure becoming pervasively available, a community comprising of leaders of major research laboratories has called for global research collaboration on one-to-one technology learning. Seamless Learning framework proposed a continuity of learning experience across different environments (Chan et al., 2006). Under SLE, students can continue to learn individually and within a group at any place and any time. The ENLACE project (Verdejo, Celerio, Lorenzo & Sastre, 2006) explored the use of mobile technologies to support the flow of learning activities from one scenario to another. Moving to another scenario may occur in various locations, using the physical space as a resource for learning through experiments and discovery outside the confines of the classroom (Milrad, Hoppe, Gottdenker & Jansen, 2004). In this paper, we propose a SLE triangle (Figure 1) to elaborate how we plan to design the curriculum, technologies, as well as the learning environment. Technology is the component that allows learners to collaborate across Time, Space, and Artifacts through the community. Technology allows learners to be engaged in learning at anytime and anywhere while connected to other community of learners. Time means student learning can happen at any time. Space can be physical spaces such as a classroom or a national reserve area for field trips; space can also be a social space when students work in groups mediated by artifacts or virtual spaces where students discuss on online forums. Community can be a study group, a class, or any groups of people such as peers, teachers and scientists who are concerned and involved in student learning. Artifacts facilitate knowledge construction, social discourse, and interaction among a community of learners (Stahl, 2000). These artifacts may be digitally produced on a mobile device and shared to the community of learners on the school's online portal after it is uploaded from a location outside the school.

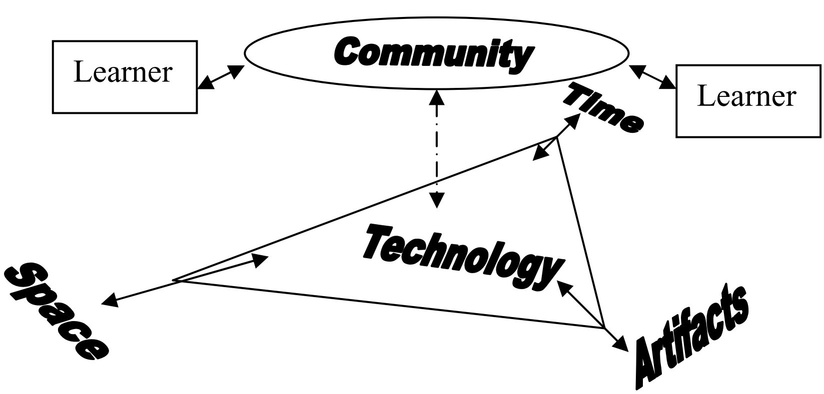


Figure 1. A framework for designing SLE

The data collected in our initial pilot suggests that a suitable lens for interpreting the learning activities is the distributed cognition theory proposed by Hollan et al. (2001) as a framework for understanding computer human interaction. They proposed three principles in which cognitive processes occur: Cognitive processes may be distributed across the members of the social group; over time; and the operation of a cognitive system in involving coordination between internal and external (material or

environmental) structure. We studied the learning activities in our recent project to understand the cognitive processes that may contribute to learning across community, time, space and artifacts. In a learning environment, technology can support cognitive processes to be part of the learning experience of the learner (van Joolingen, 1999). Studies in students' inquiry learning and discovery learning processes showed the need for scaffolding support (Krajcik, Blumenfeld, Marx, Bass & Fredericks, 1998, Jong & van Joolingen, 1998). Successful learning requires the neccesary skills and the development of cognitive processes involved in those needed skills. The existing gap between learning and the lack of needed cognitive processes are hooks for integrating intelligent suppport into design of mobile learning environments. For example, students may not have the reasoning skills in analysing the information on the mobile device to make an induction. Intelligent support can be designed to scaffold the learner in the reasoning process.

3. Extending the Research

We plan to extend our research effort to follow the questions mentioned at the beginning of the paper using a design experiment framework (Brown, 1992) with schools this year. Multiple data sources including pre-post content test, surveys of student attitude and perception change, student artifacts and online posts, interviews, classroom videos will be used to help us better understand the cognitive processes in a seamless learning environment.

References

[1] Brown, A. L. (1992). Design experiments: theoretical and methodological challenges in creating complex interventions in classroom settings. Journal of the Learning Sciences, 2(2), 141-178.

[2] Chan, T.-w., Roschelle, J., Hsi, S., Kinshuk, S., M., Brown, T., Patton, C., Cherniavsky, J., Pea, R., Norris, C., Soloway, E., Balacheff, N., Scardamalia, M., Dillenbourg, P., Looi, C., Milrad, M., & Hoppe, U. (2006). One-to-one technology-enhanced learning: An opportunity for global research collaboration. Research and Practice of Technology Enhanced Learning, 1(1), 3–29. Retrieved from http://www.worldscinet.com/rptel/01/preserved-docs/0101/S1793206806000032.pdf.

[4] Hollan, J., Hutchins, E., & Kirsch, D., (2001). Distributed Cognition: Toward a New Foundation for Human-Computer Interaction Research. In Carroll (Ed.), Human Computer Interaction in the New Millennium (pp. 75-94). New York, ACM Press.

[3] Zhang, B., Tan, N., Looi, C. K., Seow, S. K., Oh, T. T., Chen, W., & Chung, T. M. (2006). Using mobile learning technologies for primary environmental education In Singapore schools. Paper presented at the International Science Education Conference 2006, Singapore.

[4] van Joolingen, W. (1999) Cognitive tools for discovery learning, IJAIED, 10, 385-397

[5] Jong, T., van Joolingen, W. (1998) Scientific Discovery Learning with Computer Simulations of Conceptual Domains. Review of Educational Research, Vol. 68, No. 2 (Summer, 1998), pp. 179-201

[6] Stahl, G. (2000) Collaborative Information environments to support knowledge construction by communities, AI Society, 14, 71-97.

[7] Krajcik. J., Blumenfeld, P., & Marx R. W. Bass, B. M, Fredericks, J. (1998) Inquiry in Project-Based Science Classrooms: Initial Attempts by Middle School Students. The Journal of the Learning Sciences, 7(3&4), pp 313-350.

[8] Milrad, M. Hoppe, U. Gottdenker, J. Jansen, M. (2004). Exploring the use of mobile devices to facilitate educational interoperability around digitally enhanced experiments, WMTE'2004, pp. 182-186.

[9] Verdejo, M. F., Celorrio, C., Lorenzo, E., and Sastre, T. (2006). An Educational Networking Infrastructure Supporting Ubiquitous Learning for School Students. In Proceedings of the Sixth IEEE international Conference on Advanced Learning Technologies (July 05 - 07, 2006). ICALT'2006. IEEE Computer Society, Washington, DC, 174-178.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Optimizing Student Models for Causality

Benjamin Shih, [a] Kenneth Koedinger, [a] Richard Scheines[a]
[a] *Carnegie Mellon University*

Abstract. Complex student models often include key parameters critical to their behavior and effectiveness. For example, one meta-cognitive model of student help-seeking in intelligent tutors includes 15 rules and 10 parameters. We explore whether or not this model can be improved both in accuracy and generalization by using a variety of techniques to select and tune parameters. We show that such techniques are important by demonstrating that the normal method of fitting parameters on an initial data set generalizes poorly to new test data sets. We then show that stepwise regression can improve generalization, but at a cost to initial performance. Finally, we show that causal search algorithms can yield simpler models that perform comparably on test data, but without the loss in training set performance. The resulting help-seeking model is easier to understand and classifies a more realistic number of student actions as help-seeking errors.

1. Introduction

Student models for intelligent tutoring systems often require parameters for which little research or domain expertise exists. Incorrectly chosen parameter values can then lead to ineffective interventions. Machine learning algorithms offer one solution, but the results are often not human-readable and may not generalize well[3]. Further, while most algorithms try to improve correlations, e.g. between tutor help and learning, interventions require *causal* relationships. Using a student model and log data from prior research[2], we show that directly tuning parameters gives poor generalization, but that model reduction algorithms can improve that generalization. We then show that causal search algorithms provide still better generalization, but with simpler models, better initial performance, and more robustness.

We optimize a help-seeking model defined by Aleven et. al.[2] The model is a set of production rules, where each rule is an "AND" of thresholds. A transaction is classified as a help-seeking error if at least one rule is satisfied. See Figure 1 for an example. The model has 10 thresholds and 15 rules. We use two data sets from prior research with the help-seeking model. The model was initially tested on the first data set, in which the experimental condition was asked to explain their work by choosing a justifying theorem[1]. 40 students have pre-post scores. In the second data set, the help-seeking model is used to guide help-seeking interventions[5]. There were 30 students with pre-post scores, but we only use the control group, which did not receive a help-seeking intervention[5]. We will refer to these two data sets as "02" and "06", respectively.

We consider four types of models: original, tuned, reduced, and causal. The original model comes from Aleven et. al.[2] The tuned model has parameters derived from the 02 data set. The reduced model has rules chosen by stepwise regression, while the causal model has rules chosen by a causal search algorithm. We use Greedy Equivalence Search (GES), which is provably correct in the asymptotic limit[4].

Rule: READ-HINT-QUICKLY
IF the current subgoal is to think about a hint
AND the student spent less than THRESHOLD seconds to read the hint
THEN Bug Message: "Slow down. Take some time to read the hint."

Figure 1. Example rule classifying a help-seeking error

Table 1. Original and Tuned Models

Model	Rules	02 Correlation	06 Correlation
Original	15	-0.60**	-0.07
Original + Tune	15	-0.62*	0.18

* $p < .0001$, ** $p < .00001$

Table 2. Stepwise Reduced Models

Model	Num. Rules	02 Correlation	06 Correlation
Stepwise	5	-0.13	-0.22
Stepwise + Tune	5	-0.20	-0.23
Tune + Stepwise	5	-0.53**	0.09
Tune + Stepwise + Tune	5	-0.54**	0.03

* $p < .05$, ** $p < .001$

2. Results

In the following tables, we list the number of rules in each model and the correlations on both the 02 and 06 data sets. The correlations are between the percentage of a student's transactions classified as help-seeking errors, and the student's pre-post learning gain. The models are labeled with the steps used to generate them, e.g., "Tune + Stepwise + Tune" corresponds to taking the original model, tuning its thresholds, performing a stepwise regression, and tuning the resulting model's thresholds again. The results for the original and tuned models are shown in Table 1. The original model's performance on 02 was promising ($p < .0001$), but the model does not generalize well to 06 ($r = 0.18$). Tuning the model gave similar 02 performance, but terrible generalization.

To improve generalization, we used stepwise regression. The results are shown in Table 2. Performance on the test set improved after model reduction, demonstrating that some rules were unnecessary. We then tried tuning the model before performing the stepwise regression, but the results were poor.

While stepwise regression keeps rules based on statistical significance, we ideally want to keep rules based on their causal relationship with learning. Thus, we used the GES algorithm with $\alpha = .05$, resulting in a directed acyclic graph representing the causal relationships between variables. We kept only rules directly causally linked to the learning gain. The results are shown in Table 3. Running GES on the original model gave strong 02 performance ($p < .01$), but poor 06 performance. Tuning the resulting model gave a dramatic improvement in the 06 correlation, matching the best previous results. The second approach, tuning the thresholds before running GES, gave similar results. The two best GES models are shown in Figure 2. Both models are much smaller than the original. While the original model has 15 rules, the GES models have 3 rules and 2 rules. While the original model needs 10 thresholds, the GES models need 4 thresholds and

Table 3. GES Reduced Models

Model	Num. Rules	02 Correlation	06 Correlation
GES	3	-0.47*	-0.04
GES + Tune	3	-0.48*	-0.23
Tune + GES	2	-0.48**	-0.04
Tune + GES + Tune	2	-0.49**	-0.20

* $p < .01$, ** $p < .001$

GES + Tune	Tune + GES + Tune
READ-HINT-QUICKLY	READ-HINT-QUICKLY
TRY-STEP-AFTER-HINT-HI	TRY-STEP-AFTER-HINT-LOW
GLOSSARY-AFTER-HINT-LOW	

Figure 2. Best GES Models

3 thresholds. Both models have similar rules, suggesting a degree of robustness. Also, the original model classified over 70% of all transactions as help-seeking errors[2], but intervening that frequently would overwhelm students. The two GES reduced models have error rates of just 37% and 42%.

3. Conclusions

The original help-seeking model suffered from three major problems: the complexity of the model, poor 06 performance, and classifying too many help-seeking errors. The two GES approaches improved the model on all three counts. For complexity, they reduced the number of thresholds by over 60% and the number of rules by over 80%. For 06 performance, they improved the correlation from -0.07 to between -0.20 and -0.23. For classification, they reduced the number of help-seeking errors by almost half. The GES models also appear more robust. While the evidence is not complete, we did show that GES reduced models can be simpler and more persuasive than the original model. More research is still needed, especially an experimental validation of the resulting models.

References

[1] Aleven, V. Koedinger, K.R. (2002) "An Effective Meta-cognitive Strategy: Learning by Doing and Explaining with a Computer-based Cognitive Tutor." Cognitive Science, 26(2), 147-179.

[2] Aleven, V., McLaren, B.M., Roll, I., Koedinger, K.R. (2004) "Toward tutoring help seeking - Applying cognitive modeling to meta-cognitive skills." Proceedings of 7th Conference on Intelligent Tutoring Systems, 227-39.

[3] Baker, R.S., Corbett, A.T., Koedinger, K.R. (2004) "Detecting Student Misuse of Intelligent Tutoring Systems." Proceedings of the 7th International Conference on Intelligent Tutoring Systems, 531-540.

[4] Chickering, D. M. (2002) "Optimal Structure Identification with Greedy Search." Journal of Machine Learning Research, 3:507-554.

[5] Roll, I., Aleven, V., McLaren, B.M., Ryu, E., Baker, R.S., Koedinger, K.R. (2006) "The Help Tutor: Does Metacognitive Feedback Improves Students' Help-Seeking Actions, Skills and Learning?" Proceedings of 8th International Conference on Intelligent Tutoring Systems, 360-9.

Sun Seeking: Interactive Story-Reading Through Different Media

Hilary SMITH[a], Rose LUCKIN[b], Danaë Stanton FRASER[c], Lisa WILLIAMS[c], Andreas DUENSER[d], Eva HORNECKER[d], Adrian WOOLARD[e], Brendan LANCASTER[e]
[a] *Human Centred Technology Group, Informatics, University of Sussex, UK*
[b]*London Knowledge Lab, Institute of Education, University of London, UK*
[c]*Department of Psychology, University of Bath, UK*
[d]*HIT Lab NZ, University of Canterbury, NZ;* [e]*BBC, London, UK*

Abstract. This poster introduces two pilot studies on children's story reading and re-telling using interactive media that move beyond familiar keyboard and mouse interaction. These studies formed a subset of investigations into the potential for Augmented Reality (AR) story books to engage and motivate story reading using familiar and easy to set-up technologies. We contrasted AR with two other media and report initial findings that point towards the memorability and potential for engagement of these different media, concluding with our further research questions.

Keywords. AR, interactive reading, informal learning, qualitative evaluation

Introduction

Ubiquitous, interactive and mobile computing has enabled learning activities to cover wider contexts of use, moving beyond traditional desktop PC interaction. Augmented Reality (AR) provides overlays of virtual images onto reality. In contrast to applications that create purely digital experiences, our AR set-up allows the user to see themselves on screen in a composite image as if present in a virtual world whilst interacting with virtual characters in front of them (Figure 1a). This is achieved through a webcam, a PC and black and white print patterns. The patterns, held in the hands and detected by the software through the webcam, are translated into animation sequences. AR can develop collaborative learning behaviours through literature, e.g. the MagicBook [5]; or cross-generational games e.g. Curball [3].

The value added by AR for children's storytelling when compared to ordinary text and text plus pop-up illustration was demonstrated by McKenzie et al. [5]. They argued that AR enhances the experience in a number of ways, e.g. by allowing the reader to be actively engaged in object manipulation the books are no longer static sources of information. Through enhanced interactive books AR can enable children to experience virtual environments, some of which could not be experienced first-hand [1, 6].

To probe deeper into the learning potential of AR, the pilot studies reported here are a follow-on set to [6, 2] by observing young children using a narrative, alongside two different types of media: Flash file and pop-up book: study A (Figure 1b, c); and by observing single and paired children in their use of the AR story only: study B. Our interest was: what kind of interaction behaviours does each medium support; what group interactions occur in the different group sizes; and which media has greater

potential for learning behaviours e.g. memorability? We now describe the studies and initial findings of our focus on one story book, ‘Looking for the Sun’ (LFTS).

Figure 1 – Story experienced as a) Augmented Reality b) Flash file and c) pop-up books

1. Studies Set-up

Two studies on the BBC-produced AR story books were conducted over summer 2006 and here we focus on the book entitled ‘Looking for the Sun’ where a group of four characters have an adventure finding the sun. On their way they explore beach picnic leftovers, try climbing on each others’ shoulders and try to catapult to the sun.

1.1. Study A’s Context and Activities Based in Brighton and Bath, UK

12 children aged 5-6 participated in their familiar classroom in groups of 4 each with a mix of strong and weak readers. They interacted with either the AR, Flash version or 2 pop-up books created by the research team using images from the Flash file and the AR file’s text. They were then individually interviewed about their story recollections.

1.2. Study B’s Context and Activities Based in Christchurch, NZ

Nine children aged 6½ to 7 participated at a library learning centre. Enthusiastic readers were chosen to interact with the media in familiar pairs (3 pairs) or as individuals (3 children). In pairs we expected more insight into the children’s thinking processes as they shared the experience. Children were interviewed after the session.

A facilitator guided all 21 children through the media, ensuring each had a chance to read the story and interact where possible. The data sources from each study were:

- A - video tapes of groups, researcher notes from 4 researchers, individual post-session interviews with each child and each child’s drawings
- B - video tapes of pairs or individuals, semi-structured post-session interviews.

2. Data Analysis and Initial Findings

We report initial findings from analysis of both sets of story recollection data from audio and video files. Each individual, pair or group’s comments about the story were transcribed and categorised by characters and story events they mentioned and drew.

2.1. Individuals and Pairs (AR only)

- Individuals had more problems with navigation and with solving tasks of interactive sequences (e.g. how to trigger required actions)

- Collaborative interaction might help children to cope with such problems by increasing the trial of more diverse interactions individually and between them
- Pairs seemed to be more playful, showed less strategic behaviour but seemed to have more problems especially with spatial confusion (resulting from the mirror view on the screen).

2.2. Groups of 4 (Across 3 Media Types)

- Children who interacted with the pop-up book remembered more items and events from the story than both the Flash and AR groups, with the Flash group remembering the least
- Items most commonly remembered were the object that the character who was catapulted landed on (Dog, Ball, Bird), the catapult, and character Scuttle.
- The AR characters represented by the black and white printed patterns were focussed on, reported and drawn more frequently than textual story events by the AR group.

3. Discussion and Future Work

The small sizes of these pilot studies do not allow us to identify more than general trends in story understanding and recall. However we found further study of the potential of AR technologies in class and at home, and across different group sizes and media types for a more thorough exploration of learning gains is warranted. A future study could test story-reproduction and memory about the media of the stories by each group again after 2 weeks and a month to determine any strong association between media type and long term memorability. We also seek to understand the relationship between engagement, interaction level and reading ability to determine which learners could benefit most from this or other kinds of AR literacy learning.

Acknowledgements

We would like to thank Little Horsted children and teachers, and our sponsors the BBC and FRST NZ (NERF).

References

[1] Billinghurst, M. (2000). Augmented Reality in Education. New Horizons for Learning, December 2002. http://www.newhorizons.org/strategies/technology/billinghurst.htm accessed 12/12/06.
[2] Kerawalla, L., Luckin, R., Woolard, A. and Seljeflot, S. (2004) Watch this space! An evaluation of using AR to teach 10-year-olds about earth, sun and moon. Report for the BBC, Sussex University.
[3] Kern, D., Stringer, M., Fitzpatrick, G. and Schmidt, A. (2006) Curball – a prototype tangible game for inter-generational play. Workshop on TICE, IEEE Press, Manchester June 2006.
[4] Kiyokawa, K., Billinghurst, M., Hayes, S.E., Gupta, A., Sannohe, Y. and Kato, H. (2002). Communication Behaviours of Co-located Users in Collaborative AR Interfaces. Proc. of ISMAR 2002.
[5] McKenzie, J. and Darnell, D. (2003) A report into augmented reality storytelling in the context of a children's workshop, New Zealand: Centre for Children's Literature and Christchurch College of Education. http://hitlabnz.org/fileman_store/2004-eyeMagic_workshop.pdf accessed 12 December 2006.
[6] Shelton, B. and Hedley, N. (2002) Using AR for teaching earth-sun relationships to undergraduate geography students. Paper presented at the 1st IEEE Int. AR Toolkit Workshop, Darmstadt, Germany.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Effect of Metacognitive Support on Student Behaviors in Learning by Teaching Environments

Jason TAN, John WAGSTER, Yanna WU, and Gautam BISWAS
Department of EECS,
Vanderbilt University, Box 1824B, Nashville, TN 37235. U.S.A.

Abstract. We have developed environments that use teaching as a metacognitive, reflective, and iterative process to help middle school students learn about complex processes. We demonstrate that metacognition and self-regulation are crucial in developing effective learners and preparing them for future learning tasks. As evidence, we discuss the impact of metacognitive support on students' learning performance and behavior patterns that promote better learning through self-monitoring.

Keywords. Metacognition, teachable agents, behavioral analysis.

Introduction

Using the learning-by-teaching paradigm, we have developed a computer system where students teach a software agent named Betty. Self-regulated learning (SRL) strategies are incorporated into Betty's persona [2]. For example, as the student teaches Betty using a concept map she occasionally responds by demonstrating reasoning through chains of events. She may query the user, and sometimes remark (right or wrong) that the answer she is deriving does not seem to make sense. This paper describes an experiment conducted in 5th grade classrooms to demonstrate the effectiveness of the TA system with metacognitive support. Specifically, we focus on the students' ability to learn and use metacognitive strategies like monitoring and debugging.

1. Experimental Study

The study was conducted in two 5th grade science classrooms in a Metro Nashville school. 54 students were divided into three groups using a stratified sampling method based on standard achievement scores in mathematics and language. Group 1 used a system where the student was taught by a computer-based pedagogical agent that emulated an *intelligent tutoring system (ITS)* [3]. This system represented our control condition to establish the differences between learning by teaching and learning by being taught. The pedagogical agent in the system, Mr. Davis, asked the students to construct a concept map to answer three sets of quiz questions. When students submitted

their maps for a quiz he provided corrective feedback that was based on errors in the quiz answers [1]. Group 2 used a basic *Learning by Teaching System (LBT)*, where students taught Betty by creating a concept map, but Betty did not demonstrate any metacognitive behaviors. The students could query Betty to see how she reasoned with what she was taught and ask her to take the quizzes. Mr. Davis graded Betty's answers, and provided the same corrective feedback as in the ITS version of the system. Group 3, the *Self Regulated Learning System (SRL),* represented our experimental condition. Like the LBT system, students taught Betty, but Betty's persona incorporated SRL behaviors. The three groups worked for seven 45-minute sessions over a period of two weeks to create their concept maps on aquatic ecosystems.

A *Preparation for Future Learning (PFL)* task was conducted approximately 12 weeks after the main study. Students taught Betty about the land-based nitrogen cycle which was a new topic they had not discussed in class. All students used a baseline system that included resource materials on the nitrogen cycle. The Mentor provided no directed feedback, and Betty did not provide metacognitive support.

2. Results and analysis

We analyzed students' understanding by evaluating the concept maps they had generated at the end of the main and PFL studies where concept map quality was determined by the number of correct concepts and links. More importantly, we analyzed students' behaviors by examining differences in behavior patterns between groups.

We found that overall the SRL group produced better concept maps than the ITS and LBT groups in both studies though not all of the differences were statistically significant. This can be attributed to a number of factors as discussed in previous studies with Betty's Brain [1][2]. Analysis of the behavioral data led us to believe that the ITS group relied mostly on the Mentor's directed feedback to debug their maps so they could pass the quiz and were therefore less prepared to learn a new domain in the PFL study. The other two groups spent more effort in developing a better understanding of the domain so they could better teach Betty. They looked beyond concepts in the quiz questions and the corrective feedback from the Mentor [2].

In order to further investigate the effect of metacognitive feedback, we analyzed the differences in behavior between groups. Student behaviors in each session of the main and PFL studies were extracted as sequences of activities from the system log files. The sequences were defined in terms of six primary activities: Edit Map (EM), Ask Query (AQ), Request Quiz (RQ), Resource Access (RA), Request Explanation (RE), and Continue Explanation (CE).

For each group we generated transition diagrams which show the probability of transitioning from one activity to another, allowing us to identify activity sequences that are characteristic of a group's overall behavior. The transition diagrams demonstrated that the EM and EM–RQ sequences dominated for the ITS group in the main study. For the LBT and SRL groups there were more EM–AQ transitions, and the SRL group is the only one that used the AQ–RE–CE sequence of activities substantially. This shows that the SRL students put in effort to debug their concept maps, before they got Betty to take the quiz. The repeated use of AQ, RE, and CE is a clear indication of their monitoring their own learning processes and that the metacognitive support of the SRL system had a positive effect on desired learning behaviors.

In order to verify the value of these activity sequence patterns, we correlated them with concept map quality in both the main and PFL study. Table 1 shows that students who exhibited the EM–RQ behavior pattern more often were the students who produced lower quality concept maps ($p < 0.01$). On the other hand, students who exhibited the EM-AQ, AQ–RE, and AQ–RE–CE patterns more produced better concept maps ($p < 0.05$). The correlations held true for the PFL study maps.

Table 1: Correlation between Behavior Patterns and Concept Map Quality

Main Study Pattern	Main Concept Map Pearson Correlation	PFL Concept Map Pearson Correlation
EM-AQ	0.48 [b]	0.22 [b]
EM-RQ	–0.36 [a]	–0.16 [b]
AQ-RE	0.29 [a]	0.38[a]
AQ-RE-CE	0.2 [b]	0.29 [a]

[a]: significant at 0.05 level; [b]: significant at 0.01 level.

In the transition diagrams for the transfer study interesting similarities in behaviors between all three groups occurred, which we will not discuss in detail here.

3. Conclusions

The results of this study provide two important pointers: (i) metacognitive support does aid in more effective learning of domain content, and (ii) metacognitive and monitoring strategies do transfer and can be very effective in helping students prepare for future learning even when they are in environments where the learning scaffolds have been removed. In terms of improving self-monitoring and metacognitive skills, the students with SRL feedback support were more likely to develop good strategies for learning in new domains. Future work will focus on domains that include dynamic processes where students learn using scientific inquiry and the learning by teaching paradigm. We hypothesize that our approach of helping students to develop cognitive strategies and better self-monitoring skills will help them develop methods for creating and refining hypotheses through observation and experiments, testing those hypotheses, and then applying them to complex problem solving tasks.

Acknowledgements: This work has been supported by a NSF ROLE Award #0231771 and a Dept. of ED IES grant #R305H060089.

References

[1] Biswas, G., D. Schwartz, K. Leelawong, N. Vye, and TAG-V, “Learning by Teaching: A New Agent Paradigm for Educational Software,” Applied Artificial Intelligence, special issue on Educational Agents, vol. 19(3), 363-392, March 2005.

[2] Tan, J., G. Biswas, & D. Schwartz, Feedback for Metacognitive Support in Learning by Teaching Environments. Feedback for Metacognitive Support in Learning by Teaching Environments, *The 28th Annual Meeting of the Cognitive Science Society*, Vancouver, Canada, 828-833, 2006.

[3] Wenger, E., *Artificial Intelligence and Tutoring Systems*. 1987, Los Altos, California: Morgan Kaufmann Publishers.

[4] Zimmerman, B. J. A Social Cognitive View of Self-Regulated Academic Learning. *Journal of Educational Psychology,* vol. 81(3), pp. 329-339, 1989.

A Multi-Dimensional Paper Recommender

Tiffany Y. TANG[1, 2] and Gordon MCCALLA[1]
[1] *ARIES Lab, Department of Computer Science, University of Saskatchewan, Canada*
{yat751, mccalla}@cs.usask.ca
[2]*Department of Computing, Hong Kong Polytechnic University, Hong Kong*
cstiffany@comp.polyu.edu.hk

Abstract. In our previous work [9, 10], we highlighted the importance of incorporating learners' pedagogical features in making paper recommendations and proposed the pedagogical-oriented paper recommender. In this paper, we report our studies in designing and evaluating a six-dimensional paper recommender. Experimental results from a human subject study of learner preferences suggest that in the e-learning domain where there are not enough papers as well as learners to start up the recommender system (RS), it is imperative to inject other factors such as the overall popularity of each paper, learner knowledge background, etc., although these factors are less important for making recommendations on movies, books, CDs.

Introduction

Recommending educational resources is not a new idea. Recker et al. [8] study the recommendation of educational resources through *Altered Vista*, where teachers and learners can submit and review comments on educational web resources. Bruskilovsky et al. [2] report a user study of *Knowledge Sea III* providing 'annotation-based' social navigation support for making personalized recommendations to students learning a computer programming language. A number of researchers have worked on making paper recommendations as well. McNee et al. [6] investigate the adoption of collaborative filtering (CF) techniques to recommend additional references for a target research paper. A similar study by Torres et al. [11] utilizes document titles and abstracts to make recommendations. Although these studies have focused on paper recommendations, unfortunately they failed to consider pedagogical factors; that is, whether or not the recommended paper is appropriate to enhance learning. To deal with this, we proposed the notion of recommending pedagogically appropriate papers [9, 10], and obtained promising results. In this paper, we will go further to propose a multi-dimensional paper recommendation technique that incorporates the pedagogical features of learners and papers to improve the quality of the recommendations.
In the next section we will present some related work on recommender systems (RSs). In section 2, we will formulate our multi-dimensional CF-based recommendation approach, with a short discussion on experimental results. We make suggestions for further research in section 3.

1. Related Work on Multi-Dimensional Recommender Systems

The majority of RSs make recommendations purely based on item-item, user-user and/or item-user correlations without considering the contextual information in which the decision making happens [e.g. 3, 4]. However, Adomavicius et al. [1] argue that dimensions of *contextual information* including *when*, *how* and *with whom* the users will consume the recommended items can directly affect users' satisfaction towards the system performance. Pazzani [7] computes the correlations between users' learning based on their demographic information. The most recent effort is a study where users' lifestyle is considered [5]. After users' lifestyle information is drawn from subjective questionnaires, the system will then compute the Pearson correlation of users' lifestyles to relate one user to another, and make predictions accordingly. Essentially, our approach is similar to those in [5, 7]; however, our context is for paper recommendation where learners' pedagogical features are used to measure the similarity between them. Furthermore, we also consider paper features in the recommendation process which is different from the existing approaches that only consider users' contextual information in making recommendations.

Specifically, we consider the following factors in our proposed multi-dimensional CFs: papers' overall-rating, popularity, value-added, degree of peer recommendation, and learners' pedagogical features such as interest and background knowledge. Due to space limits, this paper reports only some of our findings.

2. Multi-Dimensional Pedagogically-Oriented Paper Recommendations: Mechanism and Experiment Results

2.1 The Multi-Dimensional Paper Recommendation

We first consider *overall rating*, *value-addedness* and *peer recommendation* to measure the closeness of a pair of users. *Value-addedness* represents the knowledge learned from a paper from a user's perspective (in terms of ratings given by users). And *peer recommendation* represents the user's willingness to recommend a paper to other learners (also in terms of ratings by users). Since we have three different ratings (3 dimensions) for each paper, we obtain three different Pearson correlations between the ratings given by each pair of users on co-rated papers. Suppose $P_d(a, b)$ is the Pearson correlation based on the rating r_d on dimension d, where a is the target user and b is his/her neighbor. Then, we can combine those three correlations into a weighted sum Pearson correlation as:

$$P_{3D}(a,b) = w_{overall} P_{overall}(a,b) + w_{valueadd} P_{valueadd}(a,b) + w_{peer\_rec} P_{peer\_rec}(a,b) \quad (1)$$

Also, from the learner model we combine Pearson correlations of learners' interest, $P_{interest}(a, b)$, and their knowledge background, $P_{bkgrKnowledge}(a, b)$, as follows:

$$P_{2D\ StdModel}(a,b) = w_{interest} P_{interest}(a,b) + w_{bkgrKnowledge} P_{bkgrKnowledge}(a,b) \quad (2)$$

The weights on combining Pearson correlations were tuned in our experiments. After that, we combine this with $P_{3D}(a, b)$ from equation (1), getting a *5D-Pearson correlation*:

$$P_{5D}(a,b) = P_{3D}(a,b) + w_{2D} P_{2DstdModel}(a,b) \quad (3)$$

From $P_{5D}(a, b)$ we can identify a set B consisting the best n neighbors for a target user. We then use formula (4) and (5) to calculate the aggregate rating of each paper and obtain a 6D-CF based rating for the papers:

$$r_k^{5D} = \sum_B P_{5D}(a,b) r_{b,k} \quad (4)$$

$$r_k^{6D} = r_k^{5D} + w_{\tilde{r}}\ n\ \tilde{\boldsymbol{r}} \quad (5)$$

where $w_{\tilde{r}}$ is the weight of a paper's popularity $\tilde{\boldsymbol{r}}$ (the paper's average overall ratings).

Based on the ranking of r_k^{6D}, we can find the best papers to recommend to a target user. In our experiments, in addition to the 6D-CF discussed in this paper, we have also studied 3D-CFs and 4D-CFs and compared their performances.

2.2 Experiment Discussion

A study was carried out with postgraduate students with various backgrounds enrolled in a masters program at a Hong Kong university. 22 papers were selected and assigned to students as their reading assignments according to the curriculum of the course without considering the implications for our research. We received valid data from 24 students. As in our previous studies, user (learner) profiles were drawn from a questionnaire divided into four basic categories: *interest, background knowledge, job nature,* and *learning expectation*. After reading each paper, students were asked to fill in a paper feedback form where several features of each paper were to be evaluated, including *its degree of difficulty to understand, its degree of job-relatedness with the user, its interestingness, its degree of usefulness, its ability to expand the user's knowledge (value-added),* and *its overall rating*. Two of the pedagogical factors were considered: the value-added-ness and the degree of peer-recommendation, along with overall rating. Table 1 and Figure 1 show partial experimental results.

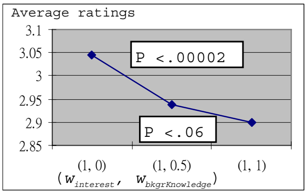
Table 1 shows the average overall ratings given by students to papers recommended by their neighbors who are determined by the technique described in section 2.1. The first column (1, 0, 0) shows the results when we use overall ratings only in finding a target user's neighbors, while the remaining columns show those when we incorporate value-added and peer-recommendation factors. It is shown here that the remaining methods yield better recommendation results in term of higher average ratings given by target users. However, the results are not statistically significant; the p-value of the t-test when the number of co-rated paper = 4 is only moderate ($p < .16$ for t-test between (1, 0, 0) and (0.4, 0.3, 0.3), and $p < .23$ for t-test between (1, 0, 0) and (0.8, 0.1, 0.1)).

Figure 1 captures the experimental results when ($w_{interest}$, $w_{bkgrKnowledge}$) = {(1, 0), (1, 0.5), (1, 1)}, w_{2D}=0.5, ($w_{overall}$, $w_{valueadd}$, w_{peer_rec}) = (100, 0, 0), and the numbers of co-rated papers are 2 (left diagram) and 4 (right

diagram). The *p*-values of adjacent data show statistically significant results. The results suggest that emphasizing knowledge background in finding a user's neigbors may lead to a less accurate recommendation.

Table 1. Average overall ratings for various compositions of ($w_{overall}$, $w_{valueadd}$, w_{peer_rec})

Number of co-rated paper	($w_{overall}$, $w_{valueadd}$, w_{peer_rec})				
	(1, 0, 0)	**(0.9, 0.05, 0.05)**	**(0.8, 0.1, 0.1)**	**(0.6, 0.2, 0.2)**	**(0.4, 0.3, 0.3)**
4	3.124	3.14	3.141	3.146	3.148
6	3.14	3.148	3.159	3.159	3.151



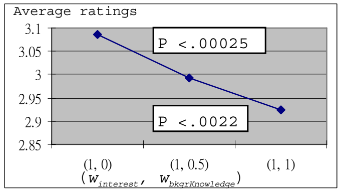


Figure 1. 6D-CF (100, 0, 0) with the number of co-rated papers as 2 (left diagram) and 4 (right diagram) in the case of recommending the best paper.

Major results suggest that *incorporating ratings from value-added-ness and peer recommendation, and considering learner interest can slightly improve the performance of CF-based RSs, especially when the number of co-rated papers is small.* However*,* we cannot accurately identify "similar" neighbors using a small number of co-rated papers. Furthermore, care should be taken when the recommender is required to pick the top 5 papers for which more information is needed if the recommender is expected to maintain its performance stability. In addition, although we did not establish a strong relationship between learners' background knowledge and the ratings, it does not necessarily mean that this relationship does not exist. In fact, in one of the papers, there are a few mathematical formulas; and a few students felt that the paper was relatively difficult to understand, even though the contents were interesting.

3. Concluding Remarks

In the presence of a limited number of co-rated papers and learners in the e-learning domain, while it is easier for a RS to pick the best paper, it is less reliable when choosing more than one quality recommended paper. This means we should look for other information extracted from both learner and paper features to help inform the RS. To accomplish this, we are studying the degree of effects that other learner features such as relatedness to their jobs and to their learning goals might have on the overall quality of recommendations.

References

[1] Adomavicius, G., Sankaranarayanan, R., Sen, S. and Tuzhilin, A. Incorporating contextual information in recommender systems using a multidimensional approach. *ACM TOIS*, 23 (1): 103-145. January 2005.
[2] Brusilovsky, P., Farzan, R. and Ahn, J. Comprehensive personalized information access in an educational digital library. *ACM/IEEE JCDL*'2005. 9-18.
[3] Herlocker, J., Konstan, J, Terveen, L. and Riedl, J. Evaluating collaborative filtering recommender systems. *ACM TOIS.* Vol.22, no. 1, 5-53, January 2004.
[4] Konstan, J., Miller, B.N., Maltz, D., Herlocker, J., Gordon, L.R. and Riedl, J. GroupLens: applying collaborative filtering to Usenet news. *CACM*, 40(3): 77-87, March 1997.
[5] Lekakos, G. and Giaglis, G. Improving the prediction accuracy of recommendation algorithms: approaches anchored on human factors. *Interacting with Computers*, 18(3): 410-431, 2006.
[6] McNee, S, Albert, I., Cosley, D., Gopalkrishnan, P., Lam, S., Rashid, A., Konstan, J. and Riedl, J. On the Recommending of Citations for Research Papers. *ACM CSCW'02*. 116-125.
[7] Pazzani, M. A Framework for collaborative, content-based, and demographic filtering. *Artificial Intelligence Rev*., 393-408, Dec. 1999.
[8] Recker, M., Walker, A. and Lawless K. What do you recommend? Implementation and analyses of collaborative information filtering of web resources for education.*Instructional Science* 31:299-316. 2003.
[9] Tang, T. Y., and McCalla, G. I. Utilizing artificial learners to help overcome the cold-start problem in a pedagogically-oriented paper recommendation system. *AH 2004*: 245-254
[10] Tang, T. Y., and McCalla, G.I. Paper annotations with learner models. *AIED 2005*: 654-661.
[11] Torres, R., McNee, S., Abel, M., Konstan, J.A. and Riedl, J. Enhancing digital libraries with TechLens. *JCDL'04*, 2004. 228-236.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Learners' Self-Organization as a First-Class Concern for CSCL Macro-Scripts

Pierre Tchounikine
LIUM, Université du Maine - France
Pierre.Tchounikine@lium.univ-lemans.fr

1. CSCL Macro-Scripts and Indirect Design

Computer Supported Collaborative Learning (CSCL) *scripts* are models designed to support collaboration among learners whose actions or interactions are (at least partially) mediated by computers. Scripts aim to enhance the probability that knowledge-generative interactions such as conflict resolution, explanation or mutual regulation occur during the collaboration process [4]. Coarse-grained scripts or *macro-scripts* orchestrate activities by setting up a given set of conditions and constraints such as sequencing the students' individual and collective tasks, creating roles within groups or constraining the mode of interaction among peers or between groups (micro-scripts are finer-grained scripts, studied at a psychological level and emphasizing individual learners' activities [2]). Technological settings designed to support CSCL scripts typically provide specific or generic communication tools (chat, mail, forum or whiteboard), awareness tools and/or tools related to the tasks, within generic (neutral) or script-specific interfaces. The intrinsic idea of CSCL scripting, that we will take as a standpoint, is that a script defines a reference frame for learners' activity.

Following the ergonomic distinction between the notions of *task* (the prescribed work) and *activity* (what people actually do), [5] emphasizes the fact that teachers set tasks and learners interpret the task, their activity being a more-or-less rational response to the task. A script is a task-related notion, based on the design by teachers of (1) the script didactic envelope [2], i.e., the pre-activities that allow triggering the script mechanisms and create favorable conditions before interaction begins; (2) the script, i.e., the way groups, roles, tasks, resources, timing or constraints are defined; (3) the technological setting, whose characteristics (functionalities, workflow) are to be studied with respect to the script, and can be used as means to influence the learners' process in a way coherent with the script. Learners' activity is related to this designed situation, but not *defined* by it. Learners' activity can not be thought of just in terms of "playing the script". Other dimensions play roles: the pedagogical and institutional context; the individual characteristics of learners; the effective motivation(s) of learners (e.g., play the game, please the teacher, solve the problem, interact with peers or gain social status) and the according effective activity/ies as related to their motivation(s); the social issues within the group such as the emergence of a leader or conflicting characters; the script and technological setting perception and appropriation by students; etc. Structuring activity is thus a challenging concept. We will thus rather consider that a script is but a factor from a set of features that influence the learner and conduct him to a given effective activity: the script is a seed and a reference, but other dimensions play a role, and macro-script enactment is unpredictable in its details.

The fact that designers have limited direct control over how their designs are enacted leads us to consider the notion of *indirect design* [6]. Within CSCL scripts, indirect design captures the idea that defining the script and the technological setting features and properties must be thought of as means to *influence* learners' activity, and that this activity (and the impact of the designed issues) must be taken into consideration as it happens, and not as it was predicted by designers. One of the core issues of CSCL macro-scripts research is to understand how to deal with these specificities and potential dilemmas, contradictions and difficulties that come hand-in-hand. In particular, the core issue is to understand (at both pedagogical and technical levels) how to influence the learners' process in a way that shapes collaborative interactions whilst avoiding over-scripting, i.e., constraining collaboration in a way that makes it sterile [3].

2. Learners' Self-Organization in Macro-Scripts

A collective phase of a script can be defined as a *collective work situation*, i.e., a situation where the multiple actors are mutually dependent in their work. Actors engaged in such interdependent processes must address an overhead activity, that of articulating (dividing, allocating, coordinating, scheduling, meshing, interrelating) their respective activities [1, 7]. This is a meta-level overhead activity that it is not focused on producing the targeted output, but on setting the conditions of the production of this output by maintaining a more-or-less stable pattern of cooperative arrangement between people engaged in the collaborative work. We will refer to this as *learners' self-organization*, "self" highlighting that, in the context of CSCL scripts, part of the organization is set by the script and part is related to learners' enactment of the script at run time (this part not necessarily being made explicit by the learners).

Learners' self-organization is a notion to be addressed as a first-class concern in CSCL macro-scripting, and its implications must be acknowledged. Within macro-scripts, the fact that learners can or have to develop some organization is correlated to the script granularity and flexibility, i.e., means and latitude that learners and teachers are presented with in order to modify some script features (e.g., groups, timing or technological setting) [2]. The way learners organize themselves impacts the learners' activity, with the script and other dimensions. Learners' self-organization is particularly related to the script as it relates to the same issue: learners' collective work structure. However, script and self-organization differ in nature: self-organization is an emergent inside-group feature; it is influenced by −but different from− the script, which is an external teacher-centred feature.

Focusing on organization, macro-scripts carry a tension between different issues: (1) a script carries constraints that define boundaries for, and impact on, learners' self-organization; (2) a script is defined by teachers; (3) a script should be appropriable by learners (scripts create a didactical contract between the teachers and the learners and between learners, and there is an assumption that learners will "play the game"); (4) the fact that people appropriate themselves a structure and/or develop a shared understanding of a structure is generally related to how much the structure has been collectively constructed and/or refined; (5) organization is a structure that emerges, is instable and evolves during activity; (6) the technological setting may impact (constrain, make impossible, allow, support) learners' activity and self-organization.

3. Acknowledging the Importance of Learners' Self-Organization

The potential role of learners' self-organization in script enactment requires this dimension to be considered when designing and experimenting the script in order, a minima, to avoid counterproductive issues and, eventually, to contribute with the script to the targeted knowledge-generative interactions. Studying learners' self-organization raises research questions such as: what relations can be drawn between self-organization issues and the learning targeted by the script and/or some other high-level skills such as autonomy or collective work skills? How can one perceive and then deal with a learners' self-organization that diverges from the script's pedagogical objectives? How to understand and deal with the dynamics of learners' self-organization, the fact that the pattern of cooperative arrangement may stabilize at some time and then be reconsidered, and the central notion of breakdown [1] which can be both an opportunity for learning and/or the cause of a collapse of the learning situation? How can learners' self-organization be impacted (influenced, supported)? How must one deal with the fact that self-organization is an activity in itself, that can be intertwined with others but may also interfere with the flow of work? We list here below some general a priori considerations.

Organization and script flexibility are two notions that are related and must be studied accordingly. In particular, the flexibility left to learners is a core issue of how learners' self-organization can emerge, and under what constraints.

The fact that some organizational issues are left open to learners may present some interest related to learners' appropriation of the script and/or learners becoming active actors in organizing their work or tackling unanticipated problems (e.g., a time management problem, a learner that downloads his contribution or an inadequate technological decision). It may also be an objective *per se*, as a means to making learners practice and learn how to work collectively.

Influencing a fundamentally emergent phenomenon such as learners' self-organization is not a self-contradiction if it is addressed in terms of indirect design and regulation and not in terms of prediction and direct design. The didactic envelope, the script structure and/or the technological setting may be used as means to provide seeds, opportunities and incentives. Teachers should have means to perceive the learners' activity and act, in particular if it appears that the emergent organization goes against the script pedagogical objectives.

Students facing a self-organization situation undertake an activity (interestingly, a collective activity). As such, it is mediated by tools from which (1) conceptual tools, that learners can use to conceptualize the cooperative work of organizing themselves and reflect on the setting and (2) technological tools, which provide functionalities that can be used by learners in the context of their self-organization process. Although not explicitly taking into account organizational issues, script settings implicitly carry potential conceptual tools (the epistemic notions used by teachers to describe the script, e.g., "group" or "role") and technological tools (the available communication or awareness tools). The fact that self-organization is usually not considered as a specific concern is widely related to the implicit basic assumption that scripts structure activity and, if any adjustment is required, learners can complement the structuring using the same conceptual notions and tools. This, however, is to be questioned: there is an issue in understanding what conceptual and technological tools can support self-organization, and if/how this support can be correlated with the script in order to influence activity in a way that is coherent (or at least not incoherent) with the pedagogical objectives.

Learners' self-organization and more generally learners' activity should be analyzed with respect to the script, but more interestingly with respect to the script pedagogical objectives, as the way these objectives have been reified in the actual script/setting may be linked to contingent issues. Scripts carry a tension between structuring the process and supporting knowledge-generative interactions. Learners' self-organization (among other issues) may encourage them to diverge from the teacher's script: is this a problem? Considering this issue, the dissociation between scripts *intrinsic* and *extrinsic* constraints (constraints bound to the script core mechanisms VS contextual factors) that has been proposed in [2] appears useful.

Finally, the script / technological setting coherence should be addressed not only by considering the script structure, but the learners' activity, which may vary (in particular because of self-organization) from the script-related previsions. Designing software to support activity is somewhat paradoxical as activity will emerge and is not fully predictable. This provides arguments for considering tailorable technological settings, i.e., provide students with means to adapt software to needs in context, according to how the script is enacted and to the underlying emergent issues such as organizational issues if any.

4. References

[1] Bardram, J. (1998). Designing for the Dynamics of Cooperative Work Activities. In Proceedings of the CSCW conference, Seattle (USA), pp 89-98.

[2] Dillenbourg, P., Tchounikine, P. (2007). Flexibility in macro-scripts for CSCL. In: Journal of Computer Assisted Learning, 23 (1), pp 1-13.

[3] Dillenbourg, P. (2002). Over-scripting CSCL: The risks of blending collaborative learning with instructional design. In P. A. Kirschner (Ed). Three worlds of CSCL. Can we support CSCL, Heerlen, Open Universiteit Nederland, pp. 61-91.

[4] Fischer, F., Mandl, H., Haake, J., Kollar, I. (2007). Scripting Computer-Supported Collaborative Learning – Cognitive, Computational, and Educational Perspectives. Computer-Supported Collaborative Learning Series, New York: Springer.

[5] Goodyear, P. (2001). Effective networked learning in higher education: notes and guidelines. Volume 3 of the Final Report to JCALT: Networked Learning in Higher Education Project. Retrieved from http://csalt.lancs.ac.uk/jisc/ (December 2nd, 2006).

[6] Jones, C., Dirckinck-Holmfeld, L., Lindström, B. (2006). A relational, indirect, meso-level approach to CSCL design in the next decade. IJCSCL 1 (1), pp 35-56.

[7] Schmidt, K., Bannon, L. (1992). Taking CSCW Seriously: Supporting Articulation Work. Computer Supported Cooperative Work International Journal 1 (1-2), pp 7-40.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Supporting Collaborative Idea Generation: A Closer Look Using Statistical Process Analysis Techniques

Hao-Chuan WANG and Carolyn P. ROSÉ
Carnegie Mellon University, 5000 Forbes Ave., Pittsburgh, PA 15213, USA
{haochuan, cprose}@cs.cmu.edu

Abstract. We are developing a conversational agent called VIBRANT to provide adaptive support for brainstorming in pairs in a scientific inquiry context. Our previous experimental study indicated that although adaptive support was effective for learning, it did not mitigate the negative influence of social interaction on idea generation. In this paper, we present a process analysis that makes the effect of the adaptive support on the idea generation process more apparent and suggests directions for future work.

Keywords. Collaborative Learning, Process Analysis, Inquiry Learning

Introduction

In this paper we explore the use of statistical process analysis techniques to examine data from an experimental study in a scientific inquiry learning setting in order to gain insights into how better to design support for productive collaborative learning interactions. Idea generation tasks related to scientific inquiry learning, simulation based learning, or project based learning are often done in groups despite overwhelming evidence of process losses during group brainstorming [2]. Learning may result from the brainstorming process itself, as it provides the impetus to engage in constructing bridging inferences similar to the process of self-explanation [1]. A significant correlation between idea generation productivity and learning in our data supports the view that learning from brainstorming may come from this constructive process [5]. Thus, an important question for collaborative learning for idea generation tasks is how to support productive idea generation and learning using adaptive collaborative learning support technology. We demonstrate how the methodology we propose can be used to gain insights such as these from corpus data.

We illustrate by example a methodology whereby a corpus of conversational interactions is first annotated with a categorical coding scheme, and how this process can be facilitated with a publicly available toolset called TagHelper tools (http://www.cs.cmu.edu/~cprose/TagHelper.html). A statistical process analysis is then conducted on top of this representation. We conclude with a discussion of lessons learned.

1. Motivation

Previously we have presented the results of an experimental study investigating the connection between brainstorming productivity and domain learning during a scientific inquiry learning task, namely the Debris Flow Hazard Task (DFH) [5]. The DFH task involves two related brainstorming tasks, one focused on *problem finding*, and the other related to *problem solving*. In the remainder of the paper we refer to these two brainstorming tasks as task 1 and task 2. The experimental study that provides a running example for this paper is a 2X2 factorial design with two between subjects factors. The first independent variable we manipulated in our experimental study was the Pair factor, i.e., whether students worked individually or in pairs. The second independent factor is referred to as Support. For this factor we manipulated whether or not students received adaptive support from the VIBRANT agent [5], which provides stimulus in the form of suggested categories of ideas [4]. A total of 42 Taiwanese high school students participated in the study. In the two conditions where students worked individually, 7 students were randomly assigned to each condition, while in the two pairs conditions, 14 students were randomly selected to form 7 pairs. The experimental manipulation took place during the first brainstorming task, which is the task related to problem finding. In addition to evaluating the success of idea generation productivity during the main brainstorming task, we also measured pre-/post-test learning gains as well as productivity on the second brainstorming task building on the earlier task, which was performed individually as a practical assessment.

The results of the experiment demonstrated a strong connection between idea generation productivity and learning. Students in the pairs condition were significantly less productive and learned significantly less

during the initial brainstorming task than students in the individual condition. On the other hand, the students who brainstormed in pairs during the first task performed better on the second brainstorming task following up on the problem finding task with a related problem solving task. Furthermore, although Support had a positive effect on learning both in the individual and pairs conditions, it did not have a significant positive effect overall on productivity during the initial brainstorming task, although there was a trend for students in the supported conditions to have a higher productivity in their idea generation. We were intrigued to find that students in the individuals with support condition, who performed best on the first brainstorming task and gain most between pre and post-test, performed worst on the second brainstorming task. The best preparation for the second brainstorming task was brainstorming in pairs with support during the first task. Thus, we were left with the mystery of how to achieve success consistently across all of our dependent measures.

The evidence suggests that working in pairs allowed students to approach the first brainstorming task more broadly, seeing it as problem finding in preparation for problem solving rather than problem finding more narrowly. There was a significant main effect of Pairs on the conditional probability that a solution was offered in the second brainstorming task given that a related problem was identified during the initial brainstorming task $F(1,38) = 4.19$, $p < .05$, $d=.6$. We also have evidence that this conditional probability, operationalizing the extent to which students conceptualized the first brainstorming task as preparation for problem solving, mediated the effect of condition on task 2 success in that there was a significant correlation between this conditional probability and task 2 success, $R^2=.49$, $p<.0001$, $N=42$. Because working in pairs with support lead to positive effects on the second brainstorming task, in this paper we take a closer look at the impact of the support provided by the VIBRANT agent on the idea generation process to better understand how it can effectively be supported with the ultimate goal of achieving high productivity on both brainstorming tasks as well as high learning gains as measured by the test.

2. Process Analysis

With our process analysis, we evaluate the effect of our experimental manipulation on the process of idea generation in connection with two important characteristics of idea generation, namely coherence [4] and the temporal characteristics of the idea generation process [2]. Specifically, by coherence we are referring to the finding that idea generation is more efficient when topic transitions between related ideas are more frequent than transitions between unrelated ideas. As for the temporal characteristics, it is conjectured that productivity losses occur mainly at the earliest stage of group idea generation. The logical consequence of this hypothesis is that the discrepancy between individual and group idea generation may vary over time. In our analysis, we make use of the discourse from our idea generation study, which has previously been coded with a set of 19 specific ideas that fit into 5 general idea topics, with a reliability of .81 Kappa [5]. This topic based analysis can be replicated automatically with high reliability (Kappa .7) using the TagHelper tools package http://www.cs.cmu.edu/~cprose/TagHelper.html.

In the first analysis, we modeled students using probability distributions that characterized their trajectory of shifting between the five general topics in the DFH domain during idea generation. Given a set of potential brainstorming ideas W, for any student s in the set of participating students S, the list of generated ideas is represented as $C(s) = <c_1,\ldots, c_{n-1}, c_n>$ where c_n denotes the general topic of the idea w that s expressed at time stamp t_n . We may then quantify the pattern of topic transitions for a later comparison across students by estimating the conditional probability of an idea being generated within a given general topic given the student and the general topic of the last idea uttered by that student. There were five general topics of "valuable ideas" in this domain, and we also considered non-valuable ideas as a separate category of ideas (i.e., ideas out of the domain model). Altogether, probabilities for 36 transitions were thus computed for each student.

We first explored the effect of our experimental manipulation on the coherence of idea generation. In order to do this analysis, we classified each topic transition as within topic or between topics. If the average of within topic transitions for condition X are higher than in condition Y, we can say topic transitions within condition X are more coherent than in condition Y. We did an analysis of variance (ANOVA) with conditional probability associated with a topic transition as a repeated dependent measure and the two independent variables from our experimental manipulation as well as the within versus between topic factor (WVBT) and Student nested within Pair and Support as independent variables. There was a significant main effect of WVBT such that transitions within topic were more frequent overall than between topics $F(1,1466) = 47.0$, $p < .0001$. More importantly, there was a significant three way interaction between Pair, WVBT, and Support $F(1,1466)= 6.3$, $p < .01$. A Tukey posthoc analysis shows that supported brainstorming in pairs is significantly more coherent than unsupported brainstorming in pairs. Brainstorming for individuals with or without support is not statistically different from the other conditions. Thus, we conclude that our brainstorming support had a positive effect on coherence, at least in the pairs condition. Infrequently students mentioned ideas during the first task that counted as solutions for the second task. These occurred mainly in the pairs with support condition. On the whole this evidence suggests that students who worked with support

were encouraged to engage with more vigor in the problem as they construed it. As discussed in the previous section, students who worked individually construed the problem more narrowly, and achieved greater success on that narrowly defined task. Considering the earlier analysis in connection with this coherence analysis, we see that students who worked in pairs construed the problem more broadly and stayed on the same topic longer, which afforded them the opportunity to pursue the problem more fully.

One might be concerned that the effect of support on coherence might mean that we must sacrifice variety in the exploration of the idea space. However, an additional statistical analysis alleviated these fears. By modeling students as probability distributions, we can compare students to one another based on Kullback-Leibler divergence [3]. For two students with probability vectors PV_1 and PV_2, a symmetric distance measure of how similar their behaviour is can be derived by averaging two values $D_{KL}(PV_1 \| PV_2)$ and $D_{KL}(PV_2 \| PV_1)$. For every student, we computed the average distance between her/him and other two types of students—either in the *same* experimental condition or in all of the *different* ones. The two measures in effect captured the information of within-condition and between-condition divergence in terms of topic transitions. A greater average distance measure in a specific condition can be interpreted to mean that condition encourages more variety in idea generation patterns. A repeated measures ANOVA analysis was performed to compare the homogeneity of topic transition patterns across experimental conditions and types of measures (i.e., "Same" or "Different"). Our independent variables included Pair, Support, Same versus Different (SVD), and Student nested within Pair and Support. Most importantly, there was a significant main effect of Support, such that adaptive support enhanced variety $F(1,38) = 177.7$, $p < .0001$.

Now we extend our process analysis in order to look at how idea generation productivity changes over time. To achieve a fair comparison between individual idea generation and group idea generation, it is standard to form "nominal pairs" by randomly selecting people from experimental conditions in which participants worked individually, and then pool their ideas collectively for a comparison with real pairs [2]. We examined the effect of our experimental manipulation at different time points during the 30-minute long brainstorming session. The first five minutes of brainstorming were the most intense, and roughly half of the ideas contributed were contributed during that span of time. Up through the first 5 minute time point, a significant evidence of process loss was found ($F(1,24)=12.22$, $p<.005$), with an effect size of 1 standard deviation. During this time we do not see a positive effect of Support, and in fact see evidence of the opposite, that Support from the VIBRANT agent may have interfered with productivity during that first segment of brainstorming. However, if we then look just at brainstorming behavior after the five minute mark, we still see significant process losses due to working in pairs, but the effect size is smaller, specifically .61 ($F(1,24)=4.61$, $p<.05$). Furthermore, we find a significant main effect in favor of the Support condition, $F(1,24)= 16.43$, $p<.0005$, effect size = 1.37 s.d. Moreover, by comparing nominal pairs who received no support and real pairs who received support after the first 5 minutes, not only is there no significant disadvantage to working in pairs, we see that there is a trend in favor of supported group brainstorming. Thus, we find that support is indeed effective for mitigating process losses, just not within the first 5 minutes.

3. Conclusions

A summative evaluation of our experimental study left us with a mystery to solve, namely the question of why the condition in which students achieved the greatest success in connection with task 1 productivity and with learning was the condition where they performed worst on task 2. A statistical process analysis building upon a categorical coding of the conversational behavior revealed insights that offer us suggestions for moving ahead and gaining success consistently on all of the dependent measures we have discussed. In subsequent studies we plan to require students in all conditions to brainstorm individually for the first five minutes with no support and only then to work in pairs with support for the duration of the first task.

This work was funded in part by National Science Foundation grant number SBE0354420.

References

[1] Chi, M. H., De Leeuw, N., Chiu, M., Lavancher, C. (1994). Eliciting self-explanations improves understanding. Cognitive Science, 18, 439-488.
[2] Diehl, M. & Stroebe, W. (1991). Productivity loss in idea-generating groups: tracking down the blocking effect. Journal of Personality and Social Psychology, 61(3), 392-403.
[3] Kullback, S., and Leibler, R. A. (1951). On information and sufficiency, Annals of Mathematical Statistics 22: 79-86.
[4] Nijstad, B. A., & Stroebe, W. (2006). How the group affects the mind: a cognitive model of idea generation in groups. Personality and Social Psychology Review, 10(3), 186-213.
[5] Wang, H., Rosé, C. P., Cui, Y. Chang, C., Huang, C., & Li, T. (to appear). Thinking Hard Together: The Long and Short of Collaborative Idea Generation, Computer Supported Collaborative Learning Confernece 2007 (CSCL 2007).

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Context Based Classification for Automatic Collaborative Learning Process Analysis

Yi-Chia WANG[1], Mahesh JOSHI[1], Carolyn ROSÉ[1], Frank FISCHER[2], Armin WEINBERGER[2], Karsten STEGMANN[2]
[1]*Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh PA, 15213 USA*
{yichiaw,maheshj,cp3a}@andrew.cmu.edu
[2]*Ludwig Maximillians Universitaet, Leopoldstrasse 13, 80802 Munich, Germany*
{Frank.Fischer,Armin.Weinberger,Karsten.Stegmann}@psy.lmu.de

Abstract. We present a publicly available tool called TagHelper that can be used to support the analysis of conversational data using automatic text classification technology. The contribution of this paper is to explore the limitations of the current simple approach to text processing employed by TagHelper tools with respect to identifying context-sensitive categories of conversational behavior. TagHelper can be downloaded from http://www.cs.cmu.edu/~cprose/TagHelper.html.

Keywords. Collabortive Learning, Automatic Process Analysis

Introduction

Building on earlier work [1], we present a publicly available tool called TagHelper tools that can be used to automate the analysis of conversational data using machine learning technology. Its goal is to facilitate the process of investigating which conversational processes are effective for supporting learning. Conversational process analysis is an especially important area in the field of Computer Supported Collaborative Learning (a review of this area of research is provided in an earlier publications [1,2]). Ultimately, beyond facilitating the study of learning processes in human tutoring and collaborative learning, automatic process analyses can also be used to trigger interventions during learning interactions.

We begin by describing the functionality that TagHelper tools makes available to researchers. We then discuss some limitations of the current version of TagHelper tools and how we are addressing them in our ongoing research. We conclude with some lessons learned, which can by applied by other researchers exploring the use of machine learning technology for supporting analysis of conversational processes in a learning context.

The research contribution of this paper is to explore the limitations of the current approach to text processing employed by TagHelper tools specifically with respect to identifying context-sensitive categories of conversational behavior, which are important for characterizing ongoing interactions between two or more people. In other words, a span of text may be ambiguous with respect to its conversational function when taken out of context. If each span of text is processed in isolation, the information required to disambiguate the function of such spans of text is not available. We present a series of investigations using a corpus of newsgroup style conversational interactions collected during a series of computer supported collaborative learning studies, which has been annotated with the coding scheme described in [2]. We describe our work in progress towards using context to more robustly identify context sensitive categories of conversational behavior.

1. TagHelper Tools

TagHelper's basic classification functionality is now available to researchers to use in their own work. Because we have observed the popular usage of Microsoft Excel as an environment in which behavioral researchers commonly do their corpus analysis work, we have adopted that as our standard file format. The TagHelper application and documentation can now be downloaded free of charge from http://www.cs.cmu.edu/~cprose/TagHelper.html. Note that this version of TagHelper that is publicly available is designed to work with presegmented English, German, or Chinese texts. In order to make TagHelper tools accessible to the widest possible user base, we have set up its default behavior in such a way that users are only required to provide examples of annotated data along with un-annotated data. At the click of a button, TagHelper tools builds a model based on the annotated examples that it can then apply to the un-

annotated examples. More advanced users can experiment with a variety of available settings, which may allow them to achieve better performance with TagHelper tools. An analyst has two types of options in customizing the behavior of TagHelper. One is to manipulate the structured representation of the text, and the other is to manipulate the selection of the machine learning algorithm. These two choices are not entirely independent of one another. An insightful machine learning practitioner will think about how the representation of their data will interact with the properties of the algorithm they select. Interested readers are encouraged to read Witten & Frank's (2005) book, which provides a comprehensive introduction to the practical side of the field of machine learning. In the near future we hope to provide an on-line training course to interested practitioners.

2. Context Based Approach

Here we discuss our investigations for extending the capabilities of TagHelper tools with respect to coding schemes that rely on judgments about how spans of text relate to the context in which they appear. In our coding methodology, we first segment a message into spans of text referred to as epistemic units [2]. Each of these units of text is then assigned one code from each of seven dimensions in our multi-dimensional coding scheme. In this paper we focus on a single one of these dimensions referred to as the Social Modes of Co-Construction, which is highly context sensitive. This dimension indicates to what degree or in what ways learners refer to the contributions of their learning partners. In this dimension there are five types of social modes, namely externalizations, elicitation, quick consensus building, integration building, and conflict-oriented consensus building. There is also a category for "other" contributions. Thus, with respect to the Social Modes of Co-Construction, each message potentially has a sequence of several codes, one for each unit of text. Furthermore, using the natural structure of a threaded discussion, we can automatically extract some features that provide some clues about the discourse function of a span of text. Thus, there are two sources of context information in the structured data that we have that we make use of. First, there is the course grained thread structure, with parent-child relationships between messages. And secondly, there is the sequence of codes that are assigned to units of text within a message.

We refer to features that are related to the threaded structure of the message board as *thread structure features*. The simplest such feature is a number indicating the depth in the thread where a message appears, which is called *deep*. This is expected to improve performance somewhat since some codes within the coding scheme never appear in thread initial messages. Other context oriented features related to the thread structure are derived from relationships between spans of text appearing in the parent and child messages. One such feature indicates how semantically related a span of text is to the spans of text in the parent message. This is computed using the cosine distance between the vector representation of the span of text and that of each of the spans of text in the parent message. The smallest such distance is included as a feature.

Next we have *sequence oriented features*. We hypothesized that the sequence of codes within a message follows a semi-regular structure. In particular, the CSCL environment inserts prompts into the message buffer that students use. Students fill in text underneath these prompts. Sometimes they quote material from a previous message before inserting their own comments. We hypothesized that whether or not a piece of quoted material appears before a span of text might influence which code is appropriate. Thus, we constructed the *fsm* feature, which indicates the state of a simple automaton, which only has two states. The automaton is set to initial state (q0) at the top of a message. It makes a transition to state (q1) when it encounters a quoted span of text. Once in state (q1), the automaton remains in this state until it encounters a prompt. On encountering a prompt it makes a transition back to the initial state (q0).

The purpose of our evaluation is to contrast our proposed feature based approach with a state-of-the-art sequential learning technique [3]. Both approaches are designed to leverage context for the purpose of increasing classification accuracy on a classification task where the codes refer to the role a span of text plays in context. We first evaluate alternative combinations of features using SMO, Weka's implementation of Support Vector Machines [4]. For a sequential learning algorithm, we make use of the Collins Perceptron Learner [3]. When using the Collins Perceptron Learner, in all cases we evaluate combinations of alternative history sizes (0 and 1) and alternative feature sets (base and base+AllContext). In our experimentation we have evaluated larger history sizes as well, but the performance was consistently worse as the history size grew larger then 1. Thus, we only report results for history sizes of 0 and 1. Our evaluation demonstrates that we achieve a greater impact on performance with carefully designed, automatically extractable context oriented features.

We first evaluated the feature based approach. The results demonstrate that statistically significant improvements are achieved by adding context oriented features (See Figure 1). All pairwise contrasts between alternative feature sets are statistically significant. The results for sequential learning are much weaker than for the feature based approach. First, the Collins Perceptron learner consistently performs significantly worse that SMO with or without context features. However, even restricting our evaluation of sequential learning to

a comparison between the Collins Perceptron learner with a history of 0 (i.e., no history) with the same learner using a history of 1, we only see a statistically significant improvement on the Social Modes of Co-Construction dimension using only base features. Note that the standard deviation in the performance across folds was much higher with the Collins Perceptron learner, so that a much greater difference in average would be required in order to achieve statistical significance. Nevertheless, the trend was consistently in favor of a history of 1 over a history of 0. Thus, there is some evidence that sequential learning provides some benefit. Performance was always worse with larger history sizes than 1.

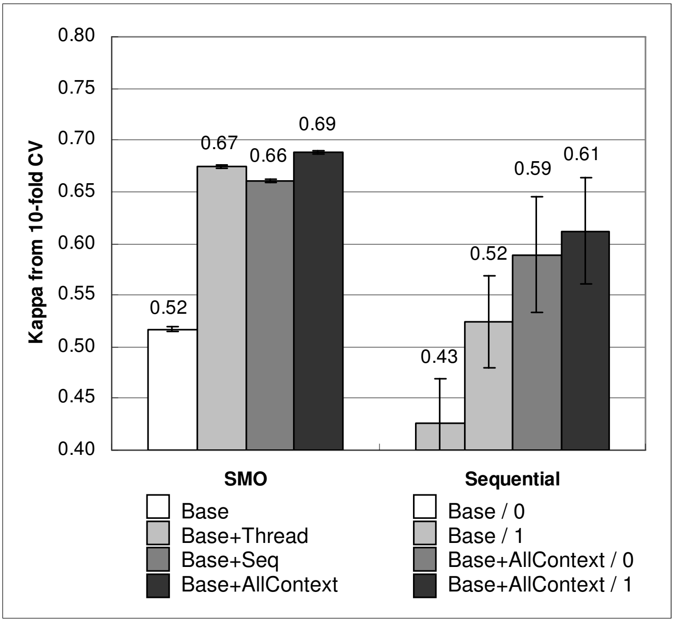


Figure 1 Performance of two classification algorithms (SMO vs Collins Perceptron) using 4 different feature sets.

3. Conclusions

We have proposed and evaluated two differed approaches for improving classification performance with a context oriented coding scheme. We showed that our selected context oriented features indeed can improve the performance of various learning algorithms, including both non-sequential and sequential ones. However, we did not observe an increase in performance due to using a sophisticated sequential learning algorithm such as the Collins Perceptron Learner. We believe the important take home message is that effectively applying machine learning to the problem of automatic collaborative learning process analysis requires selecting appropriate features that can approximate the linguistic mechanisms that underly the design of the categorical coding scheme that is used.

This project is supported by ONR Cognitive and Neural Sciences Division, Grant number N000140510043 and NSF Grant number SBE0354420.

References

[1] Donmez, P., Rose, C. P., Stegmann, K., Weinberger, A., and Fischer, F. Supporting CSCL with Automatic Corpus Analysis Technology, *Proceedings of Computer Supported Collaborative Learning* (2005).

[2] Weinberger, A., & Fischer, F. A framework to analyze argumentative knowledge construction in computer-supported collaborative learning. *Computers & Education*, 46, 71-95 (2006).

[3] Witten, I. H. & Frank, E. *Data Mining: Practical Machine Learning Tools and Techniques*, second edition, Elsevier: San Francisco, ISBN 0-12-088407-0 (2005).

[4] Michael Collins. Discriminative Training Methods for Hidden Markov Models: Theory and Experiments with Perceptron Algorithms. In *Proceedings of Empirical Methods for Natural Language Processing 2002*. (2002)

Towards a General Model for Supporting Explanations to Enhance Learning

Amali WEERASINGHE, Antonija MITROVIC and Brent MARTIN
Intelligent Computer Tutoring Group
University of Canterbury, Christchurch, New Zealand

Abstract. We present a project with the goal of developing a general model for supporting explanations, which could be used in both well- and ill-defined instructional tasks. We have previously studied how human tutors provided additional support to students learning with an existing intelligent tutoring system. Analysis of the interactions by human tutors indicates that they have helped the students to improve their understanding of database design. This paper presents the explanation model developed based on these findings.

Introduction

Even though intelligent tutoring systems (ITS) have been very effective in supporting students' learning, some students attempt to *game the system* in order to complete the problems [2]. As a result, those students acquire shallow knowledge which is sufficient only to pass exams not to solve transfer problems successfully. Self-explanation (SE) has been shown to facilitate deep learning [3]. Several ITSs were enhanced with self-explanation support in domains such as physics [4], mathematics [1], database design [8] and data normalization [6]. All these instructional tasks except database design are well-defined, as problem solving is well structured, and therefore self-explanation expected can be clearly defined. Database design is an ill-defined task: the final result is defined in abstract terms, but there is no algorithm to find it.

Our long-term goal is to develop a model of explanation which will provide adaptive support across domains. The main objective of this model is to assist students to develop their self-explanation skills while being prompted to explain their mistakes to the tutor. Since we previously implemented explanation support for the database design tutor [8], the initial work started with the same tutor. As the first step, we conducted an observational study [7] focusing on how students interacted with EER-Tutor [6], while getting additional help from a human tutor through a chat interface. A model to facilitate explanations was then developed. This model is presented in the next section followed by future work.

1. Prototype of the Self-Explanation Model

The explanation model will be used to decide when to prompt for explanations, what to explain and how to obtain explanations from learners. The model consists of three parts: error hierarchy, tutorial dialogues and rules for adapting them. Each component

is now described in turn. Error hierarchy and the dialogues are used to determine timing and content of the explanations respectively. Learners will be able to provide explanations by selecting the correct one from a list provided by the tutor.

Error Hierarchy: The domain model of constraint-based tutors is represented as a set of constraints [6]. Violations of constraints indicate mistakes in students' solutions. In previous work, we developed a hierarchy of errors students make in the Entity-Relationship (ER) domain [8], which categorizes errors as being syntactic or semantic in nature. Syntax errors are simple, each requiring only one feedback message to be given to the student; for that reason, every syntactic error corresponds to a particular constraint being violated. There were 12 such constraints. For example, constraint 8 is violated when the student creates an attribute which does not belong to an entity/relationship type. The hierarchy for semantic errors is deeper, with error types further divided into sub-errors: (i) *Using an incorrect construct type*, (ii) *Extra constructs*, (iii) *Missing constructs*, (iv) *Connecting an attribute to an incorrect construct* and (v) *Errors dealing with cardinalities and participation*. These nodes are ordered from basic domain principles to more complicated ones. Violated constraints for each type of error are represented as leaves of the hierarchy.

The ER error hierarchy was developed for a particular domain, so we were interested whether it can be reused in other domains. With that goal, we tried to fit the errors for ER-to-relational-mapping, data normalization and fraction addition. ER-to-relational mapping involves mapping an ER schema to a relational schema using the 7-step mapping algorithm [5]. Data normalization is the process of refining a relational database schema in order to ensure that all relations are of high quality. All these tasks are well-defined, due to the deterministic algorithms used.

During this investigation, several refinements were identified. The highest level of the refined error hierarchy has two nodes: *Basic syntax errors* and *Errors dealing with the main problem solving activity*. The second node is now further categorized into five nodes: (i) *Using an incorrect construct type* (ii) *Extra constructs* (iii) *Missing constructs* (iv) *Associations* and (v) *Failure to complete related changes*. The subsequent levels deal with domain-specific concepts. The common feature in all these tasks is that the syntactic and semantic accuracy of a solution can be completely evaluated by the components of the solution and its associations. However, there are exceptions. For instance, in reading and comprehension, where learners are asked to answer questions based on a paragraph, the accuracy of an answer cannot be evaluated by checking only for the correct words according to the grammatical rules. We also need to understand the implicit semantic meaning of the sentence. Therefore, our error hierarchy is not useful in such cases. In summary, we have been able to use this hierarchy in four different types of tasks: thus we believe it would be sufficiently general to be used for different types of instructional tasks only when the solution can be completely evaluated by the components of the solution and its associations.

Tutorial Dialogues: In our model, explanation is facilitated through tutorial dialogues. We present a summary here, for more details see [7]. For each error type (i.e. each leaf node in the hierarchy), we designed a dialogue consisting of four stages. In the first stage, the dialogue informs the student about the concept that s/he is having difficulty with. For example, *You seem to be having some difficulty with regular entities. Let's look at regular entities in detail. Can you tell me the general rule to decide whether something is a regular entity?* (Prompt1). The purpose of the second stage is to assist the student in understanding why the performed action is incorrect. An instance of such a prompt is *Now tell me what is unique about CHAPTER regular*

entity? (Prompt 2). The third stage prompts learners to specify how to correct the mistake. In the fourth stage, the student can review the domain concept learned. Although the prompts are domain-specific, the structure of the dialogues is domain-independent. We are currently investigating the applicability of the dialogue structure to various domains.

Rules for adapting dialogues: These rules enable individualization of the dialogues. For each student, the rules decide on the entry point into the dialogue, and/or the timing of the dialogue. Currently there are eight rules and they are based on the study [7]. For example, rule 4, dealing with customizing the entry point to the dialogue, is executed when the same error is made in the last *n* attempts. In that case, a dialogue corresponding to the mistake is initiated, but the dialogue starts from the problem-independent question (Prompt 1 above). If the error was made less than *n* attempts, then the dialogue starts from the error within the current context (Prompt 2).

Rule 1 (dealing with timing of dialogues) checks whether the student made any attempts at the current problem, and has been inactive for a specified period of time (such as 1.5 minutes, the time period we observed in the study [7]). This rule will initiate an evaluation of the student's solution even though it has not been submitted yet, and start a dialogue to discuss the most suitable error (depending on the error hierarchy and the student solution). Individualization of the chosen dialogue will be based on the rule 4 discussed above. As these rules do not depend on domain-specific details to individualise dialogues, they can be used across domains.

2. Conclusions and Future Work

Self-explanation is an effective strategy to facilitate deep learning. This research focuses on developing a model for supporting explanations for both ill-defined and well-defined tasks. This model is based on the findings of an initial study using EER-Tutor. The next step is to incorporate the model into both EER-Tutor and NORMIT. These enhanced systems will later be evaluated in authentic classroom environments.

References

1. Aleven, V., Koedinger, K. R., Cross, K. Tutoring Answer Explanation Fosters Learning with Understanding. In: Lajoie, S., Vivet, M.(eds.), Proc. AIED 1999, IOS Press, 1999, pp. 199-206.
2. Baker, R.S., Roll, I. Corbett, A.T., Koedinger, K. R. Do Performance Goals Lead Students to Game the System In: Looi, C-K., McCalla, G., Bredeweg, B., Breuker, J. (eds.) Proc. AIED 2005, IOS Press, 2005, pp. 57-64.
3. Chi, M. T. H. Self-explaining Expository Texts: The dual processes of generating inferences and repairing mental models. Advances in Instructional Psychology, 2000, 161-238.
4. Conati, C., VanLehn, K. Toward Computer-Based Support of Meta-Cognitive Skills: a Computational Framework to Coach Self-Explanation. Artificial Intelligence in Education, 11, 2000, 389-415.
5. Milik, N., Marshall, M., Mitrovic, A. Teaching Logical Database Design in ERM-Tutor. In: M. Ikeda & K. Ashley (eds). Proc. 8th Int. Conf. on Intelligent Tutoring Systems, 2006, pp. 707-709.
6. Mitrovic, A., Suraweera, P., Martin, B., Weerasinghe, A. DB-suite: Experiences with Three Intelligent, Web-based Database Tutors. Interactive Learning Research, 15(4), 2004, 409-432.
7. Weerasinghe, A., Mitrovic, A. Individualizing Self-Explanation Support for Ill-Defined Tasks in Constraint-based Tutors, Aleven, V., Ashley, K., Lynch, C. and Pinkwart, N. (eds.), Workshop on ITS for Ill-defined domains at ITS2006, 2006, pp. 56-64.
8. Weerasinghe, A., Mitrovic, A. Facilitating Deep Learning through Self-Explanation in an Open-ended Domain. Knowledge-based and Intelligent Engineering Systems, 10(1), 2006), 3-19.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Atomic Dynamic Bayesian Networks for a Responsive Student Model

Fang WEI and Glenn D. BLANK
Computer Science & Engineering, Lehigh University
19 Memorial Drive West, Bethlehem, PA 18015, USA
faw2@lehigh.edu, glennblank@gmail.com

Abstract. Atomic Dynamic Bayesian Networks (ADBNs) combine several valuable features in student models: students' performance history, prerequisite relationships, concept to solution step relationships, and student real time responsiveness. Recent work addresses some of these features but has not combined them. Such a combination is needed in an ITS that helps students learn, step by step, in a complex domain, such as object-oriented design. We evaluated ADBN-based student models 49 human students, investigating their behavior for different types of students and different slip and guess values. Holding slip and guess to equal, small values, ADBNs are able to produce accurate diagnostic rates for knowledge states over each student's learning history.

Introduction

In an ideal Intelligent Tutoring Systems (ITS), a student receives understandable and timely feedback, which not only tells how to modify the current error but also helps to rebuild his knowledge structure. Existing ITSs can flag errors, provide hints for correct solution or possible next solution step, and present definitions for relevant skills in the current problem [1][2][4][9][10]. They issue feedback based on students' errors instead of students' knowledge structure. Consequently, students oftentimes may become confused with the jungle of knowledge display and repeat the same errors, and at last quit prematurely. Ohlsson and Mitrovic [6] observe that students' knowledge acquisition requires the migration of structure from the expert's to the students'. Helping to rebuild the students' knowledge structure is an essential step in students' knowledge acquisition. However, few existing ITSs try to help the student to rebuild their knowledge structure and simulate students' knowledge in a history, because of the computational complexity of dynamic Bayesian networks tracking the volatility in the evolution of students' knowledge [5]. Our work considers both prerequisite and concept-to-solution-step relationships over a learning history, and shows how to do so in student real time, so that a student model can determine where students need help in their knowledge structure, at each step in solving a problem.

1. Atomic Dynamic Bayesian Network

An Atomic Dynamic Bayesian Network (ADBN) consists of multiple Atomic Bayesian Networks (ABN). Each ABN belongs to a single time slice when a student does a solution

step. An ABN [11] focuses on just one knowledge unit, its immediate prerequisite, and its relationship to a solution step. Based on the first-order Markov process, an ABN for the current solution step depends on the ABN of the last solution step for the same knowledge unit and not on any earlier solution steps.

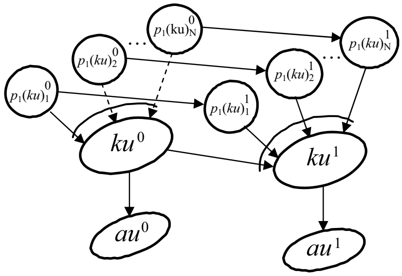


Figure 1 Atomic Dynamic Bayesian Network (ADBN) consists of ABNs from time 0 and time 1

As Figure 1 shows, an ADBN is a directed graph composed of two smaller directed graphs, in which ku^t and au^t make a pair (t=0, 1). Each immediate-prerequisite(ku) indicates a knowledge unit that must be learned before learning ku. A noisy-and relationship [3][7][11] is enforced among all edges (immediate-prerequisite(ku), ku) and edge (ku^0, ku^1). Noisy-and allows for uncertainty about a student' knowledge of ku at previous time slice and the knowledge of immediate prerequisites of ku. Similar to an ABN, all variables (nodes) in an ADBN have binary values, true or false. Causal links connect nodes for knowledge units at time slice 0 and 1. When a student understands ku at a previous time slice, he might forget ku at the current time slice. When a student does not know ku at previous time slice, he might guess the correct meaning of ku at the current time slice. These characteristics in student learning can be applied to find out the conditional tables for links between different time slices in an ADBN.

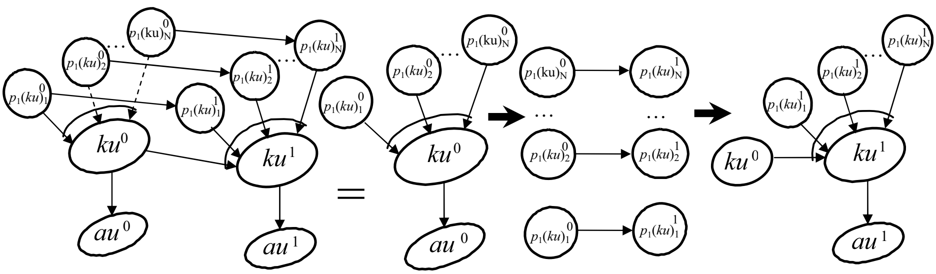


Figure 2 Update of ADBN

Figure 2 shows that an ADBN from time slices 0 and 1 is updated through three steps: The first step is to update the ABN at time slice 0, i.e., to update the center concept ku^0 and parent nodes $p_1(ku)_1^0, p_1(ku)_2^0, ..., p_1(ku)_n^0$ according to evidence au^0 at time slice1; the second step is to calculate the probability for parent nodes at time slice 1 before considering any new evidence; and the third step is to update the ABN at time slice1, i.e., to update the

center concept ku^1 and parent nodes $p_1(ku)_1^1, p_1(ku)_2^1, ..., p_1(ku)_n^1$ according to evidence au^1 at time slice 1. Step 2 simulates that a student may forget the immediate prerequisite concepts. Step 3 models that a student is very likely to understand a center concept when he understands all of the immediate prerequisites of the center concept and also knows the center concept at the previous time slice.

We evaluated the student model with 49 students using our ITS in CS1 courses at Lehigh University. We compared the diagnosed knowledge state for each student with the ones reflected from the posttest. The average correct diagnostic rate for the 49 students is 86.2%, and the average response time of ADBN after a student enters the solution step is 0.63 seconds, which is comfortably real time for the students.

2. Conclusion and Future Work

Atomic Dynamic Bayesian Networks provide an accurate student model in student real time. Each ADBN consists of two ABNs from two successive time slices. Each ABN considers a small number of immediate prerequisites, a center concept and an action step. A student model implemented with ADBNs estimate the current students' knowledge level from prerequisites in current time slice and the previous time slice. The constraints of ABN and ADBN structure and updating method guarantee real-time responsiveness.

We evaluated the sufficiency of ADBNs for student models with 49 human students. The results show that a student model using ADBNs can diagnose knowledge state of human students accurately in student real time. In future work, we will extend our student model to represent monitoring knowledge and reflective knowledge [8].

References

1. Brusilovsky, P., Schwarz, E.W., & Weber, G. (1996), ELM-ART: An Intelligent Tutoring system on World Wide Web, *Proceedings of Intelligent Tutoring Systems*, 261-269.
2. Corbett, A., Mclaughlin, M., & Scarpinatto, K.C. (2000), Modeling student knowledge: cognitive tutors in high school and college, *User modeling and user-adapted interaction*, 10, 81-108.
3. Millán, E., Agosta, J.M., Pérez-de-la-Cruz, J.L. (2001), Bayesian Student Modeling and the Problem of Parameter Specification, *British Journal of Educational Technology*, 32(2) 171-181.
4. Mitrovic, A., Mayo, M., Suraweera, P. and Martin, B. (2001), Constraint-Based Tutors: A Success Story, *Proceedings of the 14th Industrial and Engineering Applications of AI and Expert Systems Conference* (IEA/AIE-2001), Budapest, Hungary, June, 931-940.
5. Murray, W.R. (1998), A Practical Approach to Bayesian Student Modeling, *Proceedings of the 4th ITS'98 Conference*, Springer-Verlag, Berlin Heidelberg, 424-433.
6. Ohlsson, S. and Mitrovic, A. (2006), Constraint-Based Knowledge Representation for Individualized Instruction, *ComSIS*, Vol. 3, No. 1, June.
7. Peral, J. (1988), *Probabilistic Reasoning in Intelligence Systems: Networks of Plausible Inference*, San Mateo, CA: Morgan-Kaufmann
8. Self, J. (1994), Dormobile: a vehicle for metacognition, *AAI/AI-ED Technical Report*, No. 98
9. Shute, V.J. (1995), SMART Evaluation: Cognitive Diagnosis, Mastery Learning & Remediation, *Proceedings of Artificial Intelligence in Education*, 1995, 123-130.
10. VanLehn, K., Lynch, C., Schulze, K., Shapiro, J.A., Shelby, R., Taylor, L., Treacy, D., Weinstein, A., & Wintersgill, M. (2005), The Andes Physics Tutoring System: Lessons Learned, *International Journal of Artificial Intelligence and Education*, 15 (3).
11. Wei, F. & Blank, G.D. (2006), Student Modeling with Atomic Bayesian Networks, *Proceedings of the 8th International Conference on Intelligent Tutoring Systems*, ITS 2006, 491-502.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

The Distribution of Student Errors Across Schools: An Initial Study

Rob WEITZ[1], Neil HEFFERNAN[2], Viswanathan KODAGANALLUR[1] and David ROSENTHAL[1]
[1] *School of Business, Seton Hall University, South Orange, NJ 07079*
[2] *Computer. Science Department, Worcester Polytechnic Institute, Worchester, MA 01609*

Abstract. The little previous research comparing student errors across schools indicates that student "bugs" do not transfer – that is, the distribution of students' systematic errors in one school does not significantly match those in other schools. The issue has practical implications as cognitive (or "model-tracing") tutors rely on the modeling of student errors in order to provide targeted remediation. In this study we examine the responses of students at three schools to a middle-school mathematics problem. We find the same error is the most common error across all schools, and this single error accounts for some half of all incorrect responses at each school. The top five errors are similar across schools and account for some 2/3 of errors at each school. We conclude that in this example, there appears to be considerable overlap of student errors across schools.

Keywords. Student errors, cognitive tutors, constraint-based tutors, bug rules

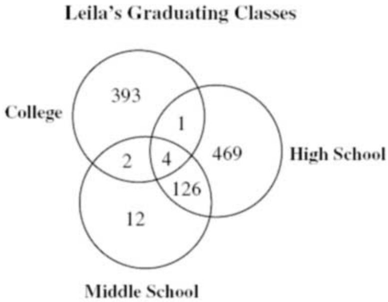
Introduction

The study of student errors in problem solving has been an active area of research inquiry in the fields of cognitive science, artificial intelligence and instructional technology for decades. The little previous research done comparing student errors across schools indicates that student "bugs" do not transfer – that is, the distribution of students' systematic errors in one school does not significantly match those in other schools.

The idea that students in different schools might make substantially different errors on the same problems is interesting in and of itself. From a practical perspective, the transferability of errors has significant importance in the area of intelligent tutoring systems. The principal method by which cognitive or model-tracing tutors [1] attempt to identify student errors is via "bug" or "buggy" rules – that is, rules that capture expected, systematic student errors in problem solving. As building cognitive tutors is a resource intensive enterprise, it would be beneficial if a tutor built based on the behavior of students at one institution could be implemented with little or no modification at other schools. Indeed, the proponents of constraint-based modeling tutors (CBMTs) claim a distinct relative advantage for their approach as, according to them, bug rules don't transfer well and CBMTs provide good quality remediation without the need for bug rules [2]. The literature seems to indicate a single study devoted to the study of the transfer of student errors across institutions. Payne and Squibb [3] examined the errors made by 13-14 year-old students on algebra problems at three (English) secondary schools. One conclusion they drew (p. 455) is that, "the rules that do the most explanatory work in the three separate groups have surprisingly little overlap."

1. The Study

In this initial study we analyzed student responses to a single question from the Assistment e-learning and e-assessing system [4]. The question is provided below in figure 1.



The Venn diagram shows Leila's graduating classes from middle school, high school, and college. How many students graduated together from **both** Leila's middle school and high school?

Figure 1: The Question Used In This Study

From the available data, we selected three schools from which there were more than 100 responses – that is, more than 100 students in each school had attempted the question. The three schools may be described as urban; they are all in the same general area of the same state. Additional data was available from three other schools that had fewer responses; additionally there were 418 responses that were inadvertently not associated with any school. We report the 10 most frequently occurring student responses from the three schools, and from all the data, in table 1 below.

2. Discussion

A striking result is that in each school, and then across all the data, the most common student answer, 126, is both incorrect and provided by roughly 35% of the students. (The correct answer is 130.) Put another way, of the students who get this question wrong, approximately half get it wrong this way, at each school. Further we see that schools 1 and 2, as well as the data for all observations, share the same top five most occurring responses, though in slightly different orders. These five errors account for some 2/3 of the incorrect responses in school 1, school 2 and across all the data. Four of these five errors appear among the top five for school 3, and here these four also comprise approximately 2/3 of the incorrect student responses. We get similar results using the top five errors of school 3.

The distribution of student errors is highly positively skewed – that is there are many rarely occurring errors. Overlap of responses across schools seems to break down after the sixth response. The skewness of the responses concurs with previous work in this area [3, 5].

3. Conclusions and Future Work

In this case, building a cognitive tutor that recognizes the most commonly occurring error at one school allows for targeted remediation for roughly 50% of the students who get the problem wrong at any school. Capturing the top five errors at one school accounts for 66% of errors at any of them. It appears that in this case, in a practical sense, bug rules do transfer across schools. More work is needed, on more problems, across more schools, and in more domains in order to confirm these results and to explore what characteristics of the domain influence the transferability of student errors.

Table 1: Top Ten Most Frequently Occurring Student Responses

School 1				School 2			
Response	**No. Students**	**%**	**Cum. %**	**Response**	**No. Students**	**%**	**Cum. %**
126	54	38.85%	38.85%	126	110	34.06%	34.06%
130	17	12.23%	51.08%	130	87	26.93%	60.99%
607	11	7.91%	58.99%	607	23	7.12%	68.11%
611	5	3.60%	62.59%	614	20	6.19%	74.30%
481	5	3.60%	66.19%	611	8	2.48%	76.78%
614	3	2.16%	68.35%	481	7	2.17%	78.95%
4	3	2.16%	70.50%	4	6	1.86%	80.80%
471	3	2.16%	72.66%	1007	5	1.55%	82.35%
1007	2	1.44%	74.10%	133	4	1.24%	83.59%
715	2	1.44%	75.54%	612	3	0.93%	84.52%
	Total Number of students:		139		Total Number of students:		323
	Number of unique responses:		40		Number of unique responses:		53

School 3				All Schools			
Response	**No. Students**	**%**	**Cum. %**	**Response**	**No. Students**	**%**	**Cum. %**
126	36	32.73%	32.73%	126	415	39.15%	39.15%
130	31	28.18%	60.91%	130	238	22.45%	61.60%
614	9	8.18%	69.09%	607	70	6.60%	68.21%
607	7	6.36%	75.45%	614	58	5.47%	73.68%
133	6	5.45%	80.91%	481	35	3.30%	76.98%
481	4	3.64%	84.55%	611	25	2.36%	79.34%
744	2	1.82%	86.36%	4	20	1.89%	81.23%
4	2	1.82%	88.18%	133	13	1.23%	82.45%
132	2	1.82%	90.00%	1007	9	0.85%	83.30%
12732	1	0.91%	90.91%	132	9	0.85%	84.15%
	Total number of students:		110		Total number of students:		1060
	Number of unique responses:		20		Number of unique responses:		111

References

[1] Koedinger, K. and Anderson, J. (1997). Intelligent Tutoring Goes to School in the Big City. *International Journal of Artificial Intelligence in Education*, 8, 30-43.

[2] Kodaganallur, V., Weitz, R. R. and Rosenthal, D. (2006). An Assessment of Constraint-Based Tutors: A Response to Mitrovic and Ohlsson's Critique of "A Comparison of Model-Tracing and Constraint-Based Intelligent Tutoring Paradigms" *International Journal of Artificial Intelligence in Education*. 16, 291-321.

[3] Payne, S.J. & Squibb, H.R. (1990). Algebra Mal-Rules and Cognitive Accounts of Error. Cognitive Science, 14, 445-481.

[4] Feng, M., Heffernan, N.T., & Koedinger, K.R. (2006). Predicting State Test Scores Better With Intelligent Tutoring Systems: Developing Metrics To Measure Assistance Required. In Ikeda, Ashley & Chan (Eds.) *Proceedings of the 8th International. Conference. on Intelligent Tutoring Systems*. Springer-Verlag 31-40.

[5] Van Lehn, K. (1982) Bugs Are Not Enough: Empirical Studies of Bugs, Impasses and Repairs in procedural Skills, *Journal of Mathematical Behavior,* 3, 2: 3-71.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Towards Learning Knowledge Objects

Amal ZOUAQ[1] , Roger NKAMBOU[2] and Claude FRASSON[1]
[1] *University of Montreal, CP 6128, Succ. Centre-Ville, Montreal, QC, H3C3J7*
[2] *UQAM, CP 8888, Succ. Centre-Ville, Montreal, QC, H3C3P8*
{zouaq, frasson}@iro.umontreal.ca, nkambou.roger@uqam.ca

Abstract. In this paper, we present an ontology-based approach, The Knowledge Puzzle approach, that aims to exploit principles from the AIED and Intelligent Tutoring Systems fields to produce e-Learning resources more tailored to learner's needs. We present a semi-automatic process to annotate learning material from different points of view: domain, structural and pedagogical. We then use this knowledge to generate dynamically learning knowledge objects based on instructional theories.

Keywords. Ontology, Intelligent Tutoring Systems, e-Learning, Learning Knowledge Objects

Introduction

The concept of learning object (LO) and particularly the concept of the reusability of LOs is one of the main research topics in the e-learning community. In fact, many initiatives for constituting learning object repositories (LORs) have been realized to facilitate the retrieval and reuse of LOs. However, LORs suffer from their lack of adaptability to a learner model as well as from the black box structure of learning objects. We believe that concepts from the AIED and ITS fields can help to overcome these shortcomings.

The paper is organized as follows: First, we present the main conceptual points of our approach. Then we detail it by presenting the learning modelling, the adopted knowledge representation and the learning knowledge object generation and standardization. We end with a conclusion.

1. The Knowledge Puzzle Approach

We propose an ontology-based approach that aims to reduce the gap between e-Learning and ITSs by dynamically aggregating learning objects on demand and according to a competence-oriented approach. Our contribution is articulated around the following main axes:

- First, we think that a competence-based approach should be adopted to allow training adapted to particular needs. In fact, the lack of adaptability to individual learners is one of the main shortcomings of traditional e-Learning approaches.
- Second, we consider that richer knowledge representations, based on ontologies, must be adopted to reflect learning object content.

- Third, we strongly agree with the fact that the pedagogical framework of learning objects is implicit and is left to the human expert [1], thus reducing the possibility to dynamically compose interesting resources. Our goal is to enable an explicit representation of the instructional framework through the creation of Learning Knowledge Objects.
- Finally, we recognize the importance of standards in the e-Learning community. Hence, we propose to produce learning knowledge objects that are compliant with the SCORM standard.

1.1. Learner Modeling: a Competence-Oriented Approach

First of all, learning objects must be adapted to fit individual needs. We developed a Competence Ontology that describes a competence as a set of skills on domain concepts. Skills are classified, in our system, according to the Bloom taxonomy [2]. Instructional objectives or competencies must be linked to learning objects or parts of them to enable their reusability efficiently. This can be done through an effective knowledge representation.

1.2. Knowledge Representation

We developed an ontological model to represent learning object content. We used the Protégé Ontology Editor [3] and the Web Ontology Language (OWL) to express the ontologies. We also developed a content model, the Knowledge Puzzle Content Model [4], that allows to annotate learning materials according to domain, structure and pedagogy:

Domain Annotation (Domain Ontology): We use machine learning (KEA-3.0 [5]) and natural language processing (Stanford Parser[6]) to generate a domain ontology from learning objects content and thus to explicit this content through concepts and relations between them. This allows for a more focused search of learning resources based not only on metadata but on the learning object content itself.

Structural Annotation (Document Structure Ontology): This ontology describes the relevant structural assets that can be found in a learning object (such as sentences, paragraphs, sections, images, tables, figures, etc). A learning object annotation using these assets is performed through a Knowledge Extractor based on IBM's UIMA [7].

Instructional Annotation (Instructional Role Ontology): The Instructional Role Ontology models pedagogical roles (e.g. Definition, Example). This is done manually, through a Knowledge Annotator.

All these ontologies are stored into an **organizational memory (OM)**, which is a dynamic knowledge prosthesis whose content is created through a knowledge management process and exploited through a learning process. We believe that an OM represents an alternative to static learning object repositories.

2. Generating Learning Knowledge Objects

An essential part of effective learning resources is the use of a clear pedagogical framework that enables a software to understand it thus helping the reuse of these

resources. In fact, we believe that an automatic approach for learning object generation should follow instructional theories. So we developed an instructional theory ontology that models an instructional theory as a set of Instructional Steps stated in the form of rules combining assets and predefined methods. These rules express a mapping between the theory's principles and the instructional role ontology. We use SWRL (Semantic Web Rule Language) as the rule formalism.

The learning knowledge object (LKO) generation follows the following path: an instructor defines a competence in term of skills on domain concepts. Starting from this definition, the system compares the learner's knowledge with the competence requirements. Then it produces an updated competence definition based on this analysis. A learning Knowledge object is generated for each skill to be mastered and according to an instructional theory chosen by the instructor. By default, the system uses the Gagné's theory, which is represented as an instance of the instructional theory ontology. Each step of the theory is attached to a set of rules that describe the assets able to fulfill the objective of the step. The final learning knowledge object takes the form of a hierarchy of activities, with each step attached to a learning resource. In order to make the learning knowledge object compatible with e-learning standards (especially SCORM), a similar structure is generated as a SCORM Content Package with a manifest describing the learning knowledge object composition and a sharable content object for each instructional step. We use a linear sequencing strategy to teach each skill.

3. Conclusion

In this paper, we presented an approach to generate dynamically learning knowledge objects (LKOs) based on instructional theories. We also showed that we are able to export such LKOs into standard formats (SCORM). We succeeded in building a bridge between fine-grained knowledge representations like the ones in ITSs and more wide-scale knowledge representations as the ones used in e-Learning. Our next goal will be to automate the process of instructional role annotation as well as to export our LKOs into the IMS-LD specification.

References.

[1] Ullrich, C. (2004). Description of an instructional ontology and its application in web services for education, in *Proc. of Workshop on Applications of Semantic Web Technologies for E-learning*, pp. 17-23, Hiroshima.
[2] Bloom, B. S. (1956). Taxonomy of educational objectives: The classification of educational goals: Handbook I, cognitive domain, Longman, New York.
[3] Protégé Ontology Editor. Retrieved from: http://protege.stanford.edu/
[4] Zouaq, A., Nkambou, R. and Frasson, C. (2006). The Knowledge Puzzle: An Integrated Approach of Intelligent Tutoring Systems and Knowledge Management, in *Proc. of IEEE ICTAI*, pp. 575-582.
[5] Frank, E., Paynter, G.W., Witten, I..H., Gutwin, C., and Nevill-Manning, C.G. (1999). Domain-specific key phrase extraction, in *Proc. of the 16th International Joint Conference on Artificial Intelligence*, pp. 668-673, Morgan Kaufmann Publishers, San Francisco.
[6] De Marneffe, M-C., MacCartney, B. and Manning, C.D. (2006). Generating Typed Dependency Parses from Phrase Structure Parses, in *Proc. of the 5th Conf. on Language Resources and Evaluation*, Genoa.
[7] UIMA. Retrieved from: http://uima-framework.sourceforge.net/

Young Researchers' Track Abstracts

This page intentionally left blank

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Toward a Generic Cognitive Model of Knowledge Representation – A Case Study of Problem Solving in DC Electrical Circuits

Amir Abdessemed [a], Mehdi Najjar [a, b], Andre Mayers [a], Habib Hamam [b]
[a] *Department of Computer Science, University of Sherbrooke, Canada*
[b] *Faculty of Engineering, University of Moncton, Canada*

Introduction

Students in electrical engineering (EE) are required (1) to fully understand the physical processes involved, and (2) to master the methods of resolving problems and perform effective analyses. Tutoring systems can improve students' skills in both aspects, but interactive simulations explaining physical processes are currently more popular than systems focused on teaching problem solving techniques.

The work presented here is part of a research consisting in establishing a generic and cognitively sound representation model of semantic and procedural knowledge with tutoring aims. The main feature of the model is to offer a structured and easily handled semantic representation of procedural knowledge in instructional design. In this extended abstract, the method is depicted along with explanations related to semantic and procedural knowledge. The effectiveness of the representation model is then discussed.

1. The Knowledge Representation Model

1.1 Semantic knowledge

The first step in representing a new domain for teaching purposes is to clearly define the concepts that the domain includes and the relations between them. Previous versions of the formalism used the Web Ontology Language (OWL) [1] to ensure a Description Logics [2] representation. Using OWL, we faced several limitations. For example, it was impossible to reason and classify concept instances according to a given datatype. The complete list of limitations and corresponding domain-related examples are presented in the full version of the paper. The new formalism uses the Jess expert system as reasoning engine. It is attribute-based: The nature of a concept depends on the list of its properties which may change during the resolution process. The reasoning engine has to infer the existence of additional instances, attributes, relations and instances of more abstract concepts. A querying mechanism allows the

retrieval of this additional data during the resolution process. In our case, parallel and serial resistors, and special cases such as current or voltage dividers are identified, and the corresponding instances and relations are created.

1.2 Procedural knowledge

According to cognitive theories [3], we assume that every user intention is a goal; and every structured set of actions to fulfill this goal is a procedure. There are two types of procedures: Primitive procedures correspond to atomic actions on the interface. Complex procedures are defined by a script of goals to satisfy. In this sense, a solution path of a given problem is a succession of goals achieved by sets of procedures. Goals and procedures are interlinked by argument passing. The arguments are instances of concepts mentally handled by the learner. An example of a goal could be *Find_Current (Current I)*, with a possible satisfying procedure *Apply_Ohms_Law_To_Find_Current* (I, R_m, U_m). Whereas the argument represented by the current I is passed directly from the goal to the procedure, the additional arguments are the results of queries symbolizing visual identification.

In our previous work, due to reasoning limitations on the semantic level, many reasoning tasks were implemented as procedures. For example, counting occurrences of particular concepts or performing a visual identification. This had no pedagogical use and led to unnecessarily large goal scripts and complex solution graphs. In the current approach, the querying system allows an automatic handling of such tasks.

2. Discussion

The problem and its resolution process were successfully represented using concise and clear goals and procedures. Our method flexibility allows experts to represent knowledge balancing between what should be represented as procedural knowledge for tutoring purposes, and what should be automated and seen either as trivial or as visual identification. Testing with more complex disciplines of EE and other scientific domains will lead to the improvement of our method.

An important matter of further work is strategic knowledge representation. When dealing with abstract domains, the goal/procedure based model is limited when explicitly representing long-term problem solving planning and resolution strategies with pedagogical objectives. Another issue is the adaptation of our authoring tool to the new model. The tool allows domain experts to represent knowledge without being familiar with the model.

References

[1] Fournier-Vigier P et al. *A Cognitive and Logic based Model for Building Glass-boxes Learning Objects*, The International Journal of Knowledge and Learning Objects, vol. 2, 2006, 77-94.

[2] *The Description Logic Handbook: Theory, Implementation, and Applications.* Baader F. et al. Cambridge University Press, 2003.

[3] Mayers A., Lefebvre B & Frasson C. *Miace : A Human Cognitive Architecture*. SIGCUE Outlook. 27(2). 2001, 61-77.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Students' Conception of Knowledge and Its Link to Knowledge Monitoring in Ill-Structured Subjects: The Potential of Representational Tools to Support Learning

Katerina AVRAMIDES
IDEAS lab, Dept. of Informatics
University of Sussex, Falmer, Brighton BN1 9QU, UK
K.Avramides@sussex.ac.uk

When we teach, whether face-to-face or through computer systems, we attempt to communicate knowledge in a manner appropriate to our learners' conceptions. This is perhaps particularly true in designing intelligent tutoring systems where we must explicitly model both the system's and the students' knowledge. In this respect, teaching ill-structured subject matter poses a great challenge, because the concepts are ill-defined and the structure of the knowledge is more fluid. This challenge is not limited to particular domains, as most, if not all of them are ill-structured at an advanced level of study. However, in 'soft' science domains, such as psychology, even the basic concepts of the domain are ill-defined. Therefore, the problems we pose to students are ill-structured even at a novice level. The design of effective learning experiences with ill-structured problems requires a fuller understanding of the factors that affect how students approach them.

The problem space in ill-structured problems is ill-specified, there are an infinite number of operations that can be applied to reach a solution and the correctness of a solution is not absolute. Therefore, addressing ill-structured problems requires making a judgement on what constitutes a valid answer. According to Kitchener's [1] theoretical framework, this requires a level of thinking above cognition and metacognition, which she terms epistemic cognition. This level of cognitive processing "is characterised as the processes an individual invokes to monitor the epistemic nature of problems and the truth value of alternative solutions" (p.225). At the risk of simplifying a complex construct it can be said that a sophisticated perspective recognises the complex, socially constructed nature of knowledge. There is substantial evidence to suggest that the majority of students lack such an understanding [2].

This research sets out to understand how students' epistemic cognition impacts on their ability to learn with ill-structured problems. More specifically, it focuses on an essential component of successful learning which is the ability to assess the state of one's current knowledge and consequently monitor and regulate the learning process.

This is a complex process that is typically considered a metacognitive skill in current frameworks [3]. The further aim is to explore the potential of representational tools to support students in constructing an accurate conception of ill-structured subject matter.

A study was carried out within the context of higher education and, specifically, the domain of psychology. There are many theoretical issues on how we conceptualise and study both epistemic cognition and the process of monitoring one's knowledge. Thus part of this research involved exploring ways of appropriately defining and assessing these theoretical constructs. It is based on the view that the study of these processes must be situated within specific learning contexts. Eight participants were recruited from psychology courses that required them to write an essay as part of their formal assessment. Participants were required to attend two semi-structured interviews (pre-and post-essay), provide any notes, a record of literature searches and a log of the essay document recorded automatically at 15min intervals.

The main finding relates to the issue of agency in knowing and how it affects students' assessment of their knowledge. The participants in this study demonstrate an understanding of knowledge in their field as relative and derived from an analysis of theory and experimental evidence. They also report that they are not in a position to have an opinion on the subject matter or critically assess theoretical frameworks and empirical studies because their knowledge is insufficient. However, this does not appear to influence their assessment of their knowledge. They report they place trust in lecture notes and textbooks to provide a complete report of the knowledge in this area and, consequently, report their knowledge at the high end of a scale from 1 to 10. The issue here is not that they place trust in authority or that they report they are unable to critically assess knowledge themselves. It is that they do not appear to consider agency to be integral to knowing. The results are discussed in terms of the role that agency plays in conceptual understanding of the subject matter, how it might affect students' learning goals and strategies, and how the way we currently teach ill-structured subjects might contribute to this perception of knowing.

This study will form the basis for investigating the potential of representational tools to support learning of ill-structured subjects. Current representation tools are focused on subject matter where basic concepts are well-defined (e.g. Belvedere, [4]). The current research will investigate the potential of representational tools to support students in understanding ill-structured subject matter. It will focus on supporting a personal analysis of the ill-defined nature of concepts and the relationships between them. It is hypothesised that this will form the basis for a deeper conceptual understanding and more accurate knowledge assessment.

References

[1] Kitchener, K.S., *Cognition, metacognition, and epistemic cognition.* Human Development, 1983. **26**: p. 222-232.

[2] Hofer, B.K. and P.R. Pintrich, *The development of epistemological theories: Beliefs about knowledge and knowing and their relation to learning.* Review of Educational Research, 1997. **67**(1): p. 88-140.

[3] Pintrich, P.R., C.A. Wolters, and G.P. Baxter, *Assessing Metacognition and Self-Regulated Learning*. In *Issues in the Measurement of Metacognition*, G. Schraw and J. Impara, Editors. 2000, Buros Institute of Mental Measurements: Lincoln, NE.

[4] Suthers, D.D., *Towards a Systematic Study of Representational Guidance for Collaborative Learning Discourse.* Journal of Universal Computer Science, 2001. **7**(3)

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Learning Engagement: What Actions of Learners Could Best Predict It?

Mihaela COCEA
National College of Ireland,
School of Informatics, Mayor Street 1, Dublin 1, Ireland

Introduction

One important aspect of motivation is engagement. In order to learn, students need to be engaged in the learning activities. However, that does not always happen due to various factors. In classroom settings, keeping track of student's level of engagement and acting accordingly is one of teacher's tasks. In e-Learning systems, the "engagement problem" is through engagement theory [1] that emerged relatively recently, even if the term existed and was used to designated learner's focus on the activity.

Research that investigates engagement has been done most of the time from a post-hoc perspective and using self-evaluation in classroom (e.g. [2]) and e-Learning (e.g. [3]) context with the purpose to determine the educational value of a course/ institution, or to find methods to improve engagement.

Unlike these approaches, ours is focused on the learning time and is based on external indicators (the actual actions of learners). Thus, we are interesting in monitoring the learners' actions in order to intervene when the student is not engaged in learning. In our research [4] identifying the level of engagement is the first step in eliciting motivation.

1. Study Design, Results and Discussion

A study was conducted using 48 log files that registered the actions of learners using HTML Tutor, a web-based system for learning HTML [5]. The research question of our study is: What actions of learners could best predict engagement?

In order to answer this question, we investigated several aspects: (a) the frequency of each possible action; (b) the attribute ranking and (c) the level of engagement prediction results. The frequency of actions gives us information about the actions that learners do most frequently; we would expect that the same actions would be reflected in (b) and (c) as well; if that wouldn't happen, like for example, if an action with a low frequency would have a good ranking and a good predictive value, the low frequency would actually indicate that that particular action is not that relevant.

All three approaches indicate that two actions are most relevant: reading pages and taking tests. The log file analysis showed a very good prediction level: around 87% for overall prediction and 93% for disengagement prediction. The attributes used are, in the order of their importance: average time spent reading, number of pages read/accessed, number of tests taken, average time spent on taking tests, number of correctly and

number of incorrectly answered tests. Several experiments showed that predictions based on attributes related to the two actions are as good as those that include a larger number of actions available in HTML Tutor.

Comparing our results to similar previous approaches, we found common attributes used for prediction: the number of correctly answered tests were also used in [6], [7] and [8]; the number of tests was used in [7]; the number of incorrectly answered tests in [7] and [8]; the time spent reading in [6] and [7] and the time spent taking tests in [7], [6] and [8]. Thus, this fact indicates the consistency of our findings.

2. Conclusion and Implications

We presented a study conducted in order to identify the actions of learners that would best indicate their level of engagement. The results show that these actions are reading pages and taking tests. A comparison with previous approaches indicated that similar attributes were used in detecting/predicting aspects of motivation/engagement, proving the consistency of the actions identified as most valuable. The difference that adds value to our approach is that we are including the learning time, while the previous approaches are focused on quiz-type activities. As the actions found relevant for engagement are related to common tasks in e-Learning systems, a module for engagement monitoring could be "attached", being beneficial in terms of keeping track of both motivation and learning outcomes, as disengagement indicates low motivation and ineffective learning.

References

[1] Shneiderman, B.: Relate-Create-Donate: A teaching/learning philosophy for the cyber-generation, Computers & Education 31, 1 (1998), 25-39
[2] Libbey, H.P.: Measuring Student Relationships to School: Attachment, Bonding, Connectedness, and Engagement, Journal of School Health, September 2004, Vol. 74, No. 7, 274-283
[3] Lippincott, J. K.: Learning, Engagement, and Technology. In Student Engagement and Information Literacy, edited by Craig Gibson. Chicago: Association of College and Research Libraries, (2006)
[4] Cocea, M.: Assessment of motivation in online learning environments. In Proceedings of Adaptive Hypermedia and Adaptive Web-Based Sys-tems, Dublin, Ireland, June, (2006) 414-418.
[5] Weber, G., Kuhl, H.-C., Weibelzahl, S.: Developing adaptive internet based courses with the authoring system NetCoach. In Hypermedia: Openness, Structural Awareness, and Adaptivity (LNAI 2266), Springer, Berlin (2001) 226-238.
[6] Qu, L., Wang N., Johnson, W.L.: Detecting the Learner's Motivational States in an Interactive Learning Environment. In C.-K. Looi et al. (Eds.), Artificial Intelligence in Education. IOS Press (2005) 547-554
[7] De Vicente, A., Pain, H.: Informing the Detection of the Students' Motivational State: an empirical Study. In Cerri, S. A.; Gouarderes, G.; Paraguau, F: Intelligent Tutoring Systems, 6th International Conference. Berlin: Springer-Verlag., (2002) 933-943
[8] Beck, J.E.: Engagement tracing: using response times to model student disengagement. Proceedings of the 12th International Conference on Artificial Intelligence in Education, Amsterdam, (2005) 88-95.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Adaptive Generation of Feedback in a Learning Environment with Smart Objects

Björn EISEN
University of Duisburg-Essen

Introduction

VanLehn et al. [1] showed that Intelligent Tutoring Systems can positively affect students´ learning. But these effects should not only be restricted to GUI-based systems. Intelligent feedback should be directly linked to the original cause in its natural environment, especially in learning environments with Smart Objects. The example Smart Objects technology which is used for the implementation of the here indicated architecture, is introduced in [2]. To generate context-adaptive feedback the new architecture uses a modular constraint checker (MCC). Herrmann [3] extended his constraints-checking approach from state-oriented checking to the checking of processes, which is mandatory for checks in various domains like the traffic-domain.

1. Architecture

To overcome the limitations of the example systems named above and to allow for easy future research in this area an architecture, that meets the requirements of a Smart Object Environment, which allows for adaptive feedback, is needed:

- The architecture should be extensible for new Smart Object technologies and different Smart Objects technologies should be usable together in one scenario.
- The integration of various feedback systems should be supported.
- A common abstraction level for the objects to be checked is needed.
- A Smart Object-technology-independent mechanism is needed to define a domain.
- Asking for the support of various feedback systems one has to support open-ended types of feedback.
- Flexible presentation of the feedback with respect to the use of Smart Objects is needed. A question to be raised is how to give the feedback depending on the object and the group size. The feedback should be given as selective as possible.
- Objects of the modelling space should be used by several learning environments at a time to enable different learning environments to be augmented by Smart Objects and vice versa.

2. Example Applications

In real world environments the presentation of feedback can be integrated into this environment, as far as the used Smart Object technology is capable. This integration allows for keeping attention only on the artefacts/environment used, no additional feedback channel has to be established, which needs special attention. Text and URL-feedback were developed so far. The work presented here extends these standard feedbacks of MCC by Text-To-Speech (TTS), Highlighting, Presentation and Method-Invocation-feedback.

An astronomical scenario and a traffic scenario were adapted to be used within the described architecture. A third Smart Plugin component, named Smart Domain was also developed, it allows for defining a scenario at runtime. The user can define the objects of the domain himself/herself by specifying the visual appearance with the help of a picture. In the astronomical scenario the feedback is of the type TTS, so the text is displayed on the board with the help of the video projector and read by the speaker system. In the traffic scenario students act as road users in traffic situations. Therefore, Smart Objects look like different road users and can be dragged around on the board, which is covered by a projected road system. Two types of feedback are used for this scenario, the first of which is the Presentation-feedback where objects are highlighted and TSS is spoken. The other feedback can invoke parametrized methods of the checked objects. This is used to make an ignored road user blow his specific horn or bell. Due to the use of the RFID Smart Objects an action cannot be performed by the Smart Object itself, but using other Smart Object technologies, it is possible to propagate the action to the Smart Object to be performed by it. The use of other Smart Objects technologies can bring this domain to a real world application, where real road users are tagged with the Smart Objects. So a real road safety education can be enriched by checking systems.

3. Conclusion

The combination of computer-enriched real world environments and intelligent feedback systems provides for a richer experience and promises to stimulate motivation. First experiences suggest this. The proposed architecture separates these systems into different independent components. This supports recombination and reusability of all components. Future work has to concentrate on the evaluation of students' learning effects of such intelligent environments.

References

[1] VanLehn, K., Lynch, C., Shulze, K., Shapiro, J. A., Shelby, R. H., Tylor, L., Treacy, D. J., Weinstein, A., Wintersgill, M.C. (2005). The Andes physics tutoring system: Five years of evaluations, Proceedings of the 12th Conference on Artificial Intelligence in Education. Amsterdam (The Netherlands), IOS Press, pp. 678-685

[2] Jansen, M., Eisen, B., Hoppe, H.U. (2005). Enriching Classroom Scenarios with Tagged Objects, Proceedings of the 12th Conference on Artificial Intelligence in Education. Amsterdam (The Netherlands), IOS Press, pp. 836-838.

[3] Herrmann, K., Hoppe, H.U.(2005). Making an Unintelligent Checker Smarter: Creating Semantic Illusion from Syntactic Analyses, Proceedings of the 12th Conference on Artificial Intelligence in Education. Amsterdam (The Netherlands), IOS Press, pp. 282-289.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Towards promoting meta-cognition using emotive interface personas within Open-Learner Modelling Environments

Sylvie GIRARD[a]
[a]*HCI Research lab, Department of Computer Science, University of Bath, BA2 7AY BATH, United Kingdom, tel:+44122538 61 83, e-mail: s.a.girard@bath.ac.uk*

Keywords: adaptive learning environments, emotive pedagogical agents, open learner modelling, meta-cognition, evaluation tool

Introduction

The use of Open-Learner Modelling (OLM) within Intelligent Learning Environments is becoming increasingly common. OLM systems indeed have the ability to improve the involvement of learners in their own learning process whilst enhancing their learning performance [1]. Interface personas appear to have real potential in motivating and engaging the user in their learning experience, in addition to facilitating human-computer interaction [2]. Similarly, meta-cognitive processes have proven to be of great importance to learning [4].

The on-going research proposed in this paper is concerned with the creation of OLM learning environments that help learners engage in meta-cognitive skills during their learning experience. This work is firmly grounded in research in Open-Learner Modelling, interface personas and affective computing.

1. Employing *DividingQuest* to investigate meta-cognition

We aim at investigating the impacts of using interface personas expressing emotions (emotive interface personas) in the design of Open-Learner Modelling tutors on children's help-seeking and self-regulation skills development.

In this article, we first present the *DividingQuest* [3, 6], a learning environment that enables empirical studies investigating the impact of using such emotive interface personas on children's meta-cognitive awareness of their learning state, to be realised. The *DividingQuest* is a Vygotskyan intelligent learning environment [5] to be used in English primary schools by year 6 children (10-11 years old). This adventure and fantasy game has been designed as an evaluation tool and includes three representations of the child-user interface. The interfaces differ in the help system, represented either by buttons, an interface persona or an interface persona displaying emotions. The Traffic-Light[1] system has been chosen as the main metaphor of learning

[1] The Traffic-Light system is used in most English schools to represent children's performance while realizing a task or grasping a concept: green for success, orange for partially understood and red for a lack of success.

in the design of the interface personas in addition to the representation of children's achievements in the game. The learning environment facilitates studies to be undertaken as to whether the use of emotive interface personas can help children develop meta-cognitive awareness of activities such as help seeking and self-regulation. The evaluation platform also enables investigations of children's level of understanding and appropriation of their own user-model.

In order to achieve the research aims, a study has been carefully designed to evaluate the impact of using emotive interface personas following the Traffic-Light system metaphor of learning. The study will involve year 6 children from local primary schools, investigating the impacts of help system representation on children's meta-cognitive awareness of their learning state, learning achievement and engagement within the learning process. The experiment has been designed as a between-subjects longitudinal comparative study with pre and post session learning tests. It includes a control and two experimental conditions, differing in the representation of the help system. Qualitative measures will be taken through the use of the *DividingQuest* and a qualitative analysis will be undertaken assessing learning outcomes. Usability evaluations will also be undertaken in the study in order to contribute to design of usable educational software to enhance meta-cognitive processes.

2. Future research directions

Our next step is to assess the utility of the evaluation tool in schools. If the results demonstrate a development of meta-cognitive skills, we will then investigate the full potential of using the traffic-light system as a metaphor of learning. An OLM learning environments will then be constructed incorporating the traffic-light system into the user-model. Results of this research will help the creation of Intelligent Learning Environments where children have a better appropriation and understanding of their User-Model component while developing meta-cognitive skills.

References

[1] Bull, S and J Kay, (2005) A framework for designing and analysing open learner modelling. *12th International Conference on Artificial Intelligence in Education (AIED 05)* Workshop 11, Kay, J, A Lum, D Zapata-Rivera, (eds) pp 81-90. 2005.

[2] Craig S., B. Gholson, & D. Driscoll. (2002) Animated pedagogical agents in multimedia educational environment: Effects of agent properties, picture features and redundancy. *Journal of Educational Psychology*, 94(2), 2002.

[3] Girard, S. and Johnson, H. (2006). DividingQuest: Using emotive interface personas in educational software. Technical Report CSBU-2006-20, Department of Computer Science, University of Bath. ISSN 1740-9497.

[4] Luckin, R., & Hammerton, L. (2002). Getting to know me: Helping learners understand their own learning needs through meta-cognitive scaffolding. In *Proceedings of Sixth International Conference on Intelligent Tutoring Systems, ITS 2002* (pp. 759-771). Berlin: Springer.

[5] Vygotsky, L.S. (1978). *Mind in society: the development of higher psychological processes*. Cambridge, MA: Harvard University Press.

[6] The Dividing Quest Project: http://www.cs.bath.ac.uk/~sasg20/DividingQuest/TheProject/

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Towards A Virtual Teaching Assistant to Answer Questions Asked by Students in Introductory Computer Science

Cecily HEINER
School of Computing, University of Utah
3190 MEB, 50 S. Central Campus Drive, Salt Lake City, Utah 84112 USA

Introduction

Many students could benefit from an automated question answering system, but research on producing automated responses to student generated questions is remarkably sparse throughout the literature. One group of students that may be especially well suited to using a semi-automated question answering tool is found in introductory computer science. Students in introductory computer science classes have had no prior exposure to programming, so they have many questions that are poorly articulated.

1. A Virtual Teaching Assistant for Introductory Computer Science

To address the problems outlined in the introduction, I propose an intelligent question answering system called a Virtual Teaching Assistant (VTA). This VTA will mediate the question answering process between the students and the teaching staff. Interaction with the VTA will work as follows. The student will type his or her question into the student software client of the VTA system. Typing the question will force students to give some thought about the question they want to ask, and hopefully facilitate deeper learning. Questions and student source code submitted to the VTA system will appear in a teaching staff's software client of the VTA system, ordered by time of submission. A member of the teaching staff will answer the question with a text or URL response. Then, the VTA system will log the answer to a database along with the question. The answer will be displayed in the interface of the student's software client and/or e-mailed to the student.

When possible, the VTA system will short-circuit the process by providing automated answers to common questions that similar to other questions already in the system. If an automated answer is provided, the student will be prompted to indicate if the answer was adequate or if they still need help from a human TA. For example, consider the following three questions:

- How do I animate the smoke? (Q1)
- What does this compiler error mean? (Q2)
- How do I make the smoke move? (Q3)

When the system receives Q1, it has not seen any other questions, so it does not have a matching response, and it will pass Q1 to a human who will type in a response A1. When the system receives Q2, the system will compare Q2 to Q1, determine that they are not similar, and pass Q2 to a human who will type in a response A2. When the system receives Q3, the system will compare Q3 to Q1, determine that there is a potential match and try to provide an automated answer A1 without human intervention. The student who asked Q3 will then indicate if A1 was a satisfactory answer.

2. A Brief Outline of the Experiment

The thesis of this work is that information from student source code, compiler output, and educational data mining can improve partially automated student question answering above a non-trivial baseline consisting of the natural language in the questions that students ask. Happily, this idea can be explored while allowing all students to use a fully functional version of the VTA system by running the experiment as a post-hoc ablation study offline. To run the experiment, the VTA system will consider each question in the order in which it was entered into the system. The system will then decide if it matches a previously entered question or if it represents a new question category. This classification is called the evaluation classification. The evaluation classification will be compared to the original classification of the question logged in the system when the student asked the question. This original classification will have been made either by a human TA or the complete VTA system; if the original classification was made by the complete VTA system, then it was verified by the student asking the question. If the evaluation classification and the original classification match, then the system will receive credit for classifying that question correctly. If they do not match, the system will receive no credit for classifying that question. The score for each system will be the percentage of questions classified correctly.

The experiment will use the VTA system with natural language as the only input source as a non-trivial baseline. The experiment will then consider four additional conditions. In the first condition, just the natural language input and the abstract form of student code will be used. In the second condition, just the natural language and the compiler output will be used. In the third condition, just the natural language and the additional information from the educational context will be used. A fourth condition will incorporate all of the input. I hypothesize that the baseline condition of the natural language alone will be able to classify approximately half of the questions, but that the additional information will allow at least an additional fourth of the questions to be correctly classified.

Acknowledgements

Thanks to my advisor Joe Zachary, reviewers, classmates in CS 7010, participants in the 2006 SIGCSE Doctoral Consortium, and Chad Lane for feedback on early versions of this work.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Exploring Narrative as a Cognitive Aid in Educational Video Games

Alan Koenig
Arizona State University

In many ways, video games – particularly those of the action/adventure genre – are characteristically similar to situated learning environments. By embodying game characters that face various physical, social, and intellectual challenges, players learn and adopt cultural models of behavior that are consistent with their character's goals and authentic to the specific game world [1]. However, in order for this situated learning paradigm to occur, the player must first understand the context of the environment in which he is placed. Video games often employ narratives to convey this needed context. Such narratives serve to inform the player of the dramatic, historical events that comprise the back-story of the game, or to further the story during game-play after significant events occur. For *educational* video games, narrative can play an additional role – that of serving as a potential cognitive aid that players can use to tie the events and character interactions of the game (both temporally and spatially) to specific, educational constructs presented in the game [2].

The means through which narrative is incorporated into games, however, is a hotly debated topic among game scholars [3, 4]. At issue is whether game narratives should be coerced to follow embedded, pre-scripted stories, or whether the narratives should emerge un-scripted as an outcome of the player's actions. Narratologists tend to favor embedded narrative in video games, regarding them as synergistic with the overall game-playing experience. In contrast, ludologists tend to oppose the embedding of narrative into games as a means of telling stories. They argue that narrative in video games should be *emergent* in nature, arising out of the action of game-play rather than being determined ahead of time.

The goal of the current study was to address this issue by exploring the impact that embedded and emergent narratives have in educational video games that model situated learning. Of interest was how both the presence and type of narrative incorporated into a game affects the player's ability to utilize and recall educational constructs presented during game-play. To carry out the experiment, 39 undergraduate students from a large university in the southwestern United States were randomly assigned to one of three conditions in which a science-related instructional video game was played. The three conditions were: (a) embedded narrative, (b) emergent narrative, and (c) no narrative (control). Players in the embedded narrative group assumed the role of a character that was part of a pre-scripted, overarching story that progressively unfolded as the game played out. Here, the player's actions and decisions could not alter the outcome of the story, but they could affect the rate at which it proceeded. In contrast, the emergent narrative group experienced the same *initial* back-story as the embedded group, but for them, the story was free to evolve in different ways based on their actions and decisions made during game-play. Finally, in the non-narrative treatment, no back-story or context was provided to the players at all.

Each treatment made use of a 3rd person perspective, 3-D computer video game designed to teach about principles of electric circuits, motors, generators, and batteries. The game environment was designed to mimic the look and feel of a main-stream action/adventure video game. In each treatment, players were tasked with leveraging various resources in the game environment to construct make-shift electrical devices that were used to solve specific challenges posed by the game.

In addition to the treatment, each participant completed a demographic survey, an attitude survey, a pre-test and posttest, and wrote a brief paragraph describing their perception of the game's narrative. Separate one-way analyses of variance (ANOVA's) were conducted to measure differences between the treatment groups on (a) students' utilization of the presented educational content to solve tasks posed in the game; (b) the amount of instructional guidance sought during game-play; (c) students' awareness of a game narrative; and (d) students' overall satisfaction with the game. In addition, an analysis of covariance (ANCOVA) was conducted to measure differences between treatment groups on students' recall of the educational content presented in the game as captured from pretest and posttest scores.

With the exception of game narrative awareness (in which the embedded and emergent groups exhibited significantly greater awareness compared to the control group), none of the analyses revealed statistical differences between the three treatment groups. This was likely due to a lack of power, as there were less than 15 participants in each condition. However, based on observations made during the experiment, a supplemental analysis was conducted to determine learning and performance differences based on gender. Surprisingly, using an independent samples t-test, males were found to score significantly higher on the posttest, had significantly greater success in accomplishing in-game educational tasks, took significantly less time to complete the game, and exhibited significantly greater overall game satisfaction.

As the results of this study clearly indicate, this experiment needs to be re-run with a power analysis conducted to ensure that differences between the treatments can be detected, as this was the primary shortcoming of this study. However, based on the observations and findings concerning the role of gender on learning and performance, a conceptual re-design of the study is in order. A 3 (narrative type) x 2 (gender) factorial design should be employed that will allow for interactions between gender and narrative type to be observed. As the supplemental analysis revealed, gaming environments can potentially be gender biased not only in terms of entertainment preferences, but also in terms of learning and performance. Further research is needed to explore how such things as game narratives, environmental settings, semiotics, lighting, and game character personas can potentially differ across gender so that they can be addressed in the design of future educational video games.

References

[1] Gee, J. P., (2003). What video games have to teach us about learning and literacy. New York: Palgrave Macmillan.

[2] Ryan, M. L. (2003). Cognitive maps and the construction of narrative space, in *Narrative Theory and the Cognitive Sciences* (D. Herman). Stanford: CSLI Publications.

[3] Frasca, G. (1999). Ludology meets narratology: Similitude and differences between (video) games and narrative. *Ludology.org Game Theory*. [Online] Available: http://www.ludology.org/articles/ludology.htm

[4] Juul, J. (1998). *A Clash between Game and Narrative*. Paper presented at the *Digital Arts and Culture* conference, Bergen, Norway. http://www.jesperjuul.net/text/clash_between_game_and_narrative.html

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Coping with Frustration: Self-efficacy Modelling and Empathetic Companion Agents

Scott W. MCQUIGGAN
Department of Computer Science, North Carolina State University, Raleigh, NC 27695
swmcquig@ncsu.edu

Affect has begun to play an increasingly important role in intelligent tutoring systems (ITSs). Recent years have seen the emergence of work on affective student modeling [3], detecting frustration [2], devising affectively informed models of social interaction [7], and detecting student motivation [4]. All of this work seeks to increase the fidelity with which affective and motivational processes are modeled and utilized in ITSs in an effort to increase the effectiveness of tutorial interactions and, ultimately, learning.

ITSs should provide support that helps students cope with emotions such as anxiety and frustration and, if possible, increasing their tolerance for frustrating learning situations. But how can we gauge what level of frustration a student should be allowed to experience before it becomes excessive? Self-efficacy influences student persistence [1]. Thus, models of student efficacy could be used to support the detection and monitoring of student frustration tolerance and thresholds. Using human tutoring as a model, our intelligent tutors, whether in the form of companions, peers, or teachers, should respond similarly to typical human empathetic behaviors. This work proposes an approach to support students coping with frustration during learning.

Self-efficacy influences students' reasoning, their level of effort, their persistence, and how they feel; it shapes how they make choices, how much resilience they exhibit when confronted with failure, and what level of success they are likely to achieve [1]. While it has not been conclusively demonstrated, many conjecture that given two students of equal abilities, the one with higher self-efficacy is more likely to perform better than the other over time. Self-efficacy is intimately related to motivation, which controls the effort and persistence with which a student approaches a task. Because effort and persistence are themselves influenced by the belief the student has that she will be able to achieve a desired outcome [1], self-efficacy holds great promise for ITSs. Self-efficacy beliefs have a stronger correlation with desired behavioral outcomes than many other motivational constructs, and it has been recognized in educational settings that self-efficacy can predict both motivation and learning effectiveness [1]. Thus, if it were possible to enable ITSs to accurately model self-efficacy, they may be able to leverage it to increase students' academic performance.

The prospect of creating an "affect learner" that can induce empirically grounded models of affect, including models of frustration, efficacy, and empathy from observations of student interactions holds much appeal. In the proposed approach we first collect training data from student interactions in the CRYSTAL ISLAND learning environment [6]. We monitor observable attributes in the environment in addition to student physiological response. The observable attributes consist of temporal and

locational features such as the student's current location, where the student has and has not been, what the student's current goal is, and how long she has been working on it. The observable attributes also contain intentional features that monitor student progress towards learning goals. Affect information is obtained through several sources. To date we have investigated using (1) physiological response, such as heart rate and skin conductivity and (2) self-reports, in which students assess their state. In the future, we plan to enrich annotations with information obtained from analysis of video recordings of facial expressions. In runtime environments we use models of affect to monitor the same attributes observed during training to recognize student affective states.

Our proposed research plan will proceed in four stages. The first stage focuses on affect modelling in ITSs [5]. This includes the development of frameworks for modelling targeted psychological constructs: empathy, self-efficacy, and frustration. The second stage investigates the associative nature of student efficacy and student frustration thresholds and tolerance levels. In this stage we will develop a study that induces student frustration and anxiety in learning environments, discovers thresholds (i.e., how much frustration the student can bear before electing to abandon the learning task) while monitoring student efficacy through self-reports, biofeedback and expert diagnosis. The third stage will integrate the first two stages and examine strategies that can be used in learning environments to facilitate successful coping behaviors and help students achieve effective emotional states for learning. This stage will require an experiment to analyze the significance of implemented strategies. This experiment will also investigate the embodiment and role of the other agents in the learning environment, assessing the merit of deploying companions vs. peers vs. tutors and how each can deliver different learning experiences and effect student psychology in unique fashions. The fourth stage will result in a comprehensive experiment to investigate the impact of the described approach on the student's learning experience including measures of learning effectiveness, learning transfer, and student engagement.

The proposed research focuses on the development of strategies to support students coping with anxiety and frustration. The architecture operates in two modes to (1) inducing models of affect, self-efficacy, and empathy, and (2) providing runtime control of student frustration threshold assessment through monitoring student levels of self-efficacy to determine the student's abilities tolerate frustrating situations. As students approach their frustration threshold, the empathy models will be used to inform the tutorial behaviors of virtual agents (companion, peer, or pedagogical agents) in their delivery of appropriate feedback customized for the student's affective state.

References

[1] Bandura, A. (1997) *Self-efficacy: The exercise of control*. New York: Freeman.

[2] Burleson, W., and Picard, R. (2004) Affective Agents: Sustaining Motivation to Learn Through Failure and a State of Stuck. *Workshop of Social and Emotional Intelligence in Learning Environments, in conjunction with the 7th International Conf. on Intelligent Tutoring Systems*, Maceio, Alagoas, Brazil.

[3] Conati, C., and Mclaren, H. (2005) Data-driven Refinement of a Probabilistic Model of User Affect. *Tenth International Conference on User Modeling*. New York, NY, pp. 40-49.

[4] de Vicente, A., and Pain, H. (2002) Informing the detection of the students' motivational state: an empirical study. In *Proc. of the 6th Intl. Conf. on Intelligent Tutoring Systems*. New York, NY, 933-943.

[5] McQuiggan, S., and Lester, J. (2006) Diagnosing Self-Efficacy in Intelligent Tutoring Systems: An Empirical Study. In *Proc. of the 8th Intl. Conf. on Intelligent Tutoring Systems*, Jhongli, Taiwan.

[6] Mott, B., McQuiggan, S., Lee, S., Lee, S.Y., and Lester, J. (2006) Narrative-centered Environments for Guided Discovery Learning, *Workshop on Agent-Based Systems for Human Learning in conjunction with fifth International Conference on Autonomous Agents and Multi-Agent Systems*, Hakodate, Japan.

[7] Paiva, A., Dias, J., Sobral, D., Aylett, R., Woods, S., Hall, L., and Zoll, C. (2005) Learning by Feeling: Evoking Empathy with Synthetic Characters. *Applied Artificial Intelligence*, 19, 235-266.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Modeling Engagement in Educational Adaptive Hypermedia

Jabari O. PULLIAM [a], Robert SAJAN [a] and Katja HOFMANN [b]
[a] *AHAT lab, California State University, East Bay, United States*
[b] *Informatics Institute, University of Amsterdam, Netherlands*

Keywords.
adaptive hypermedia, informal education, user modeling, adaptive presentation, engagement, cognitive modeling

1. Introduction

Research has shown that increasing student engagement in an Educational Hypermedia system increases academic performance. In order to engage individual users, Educational Adaptive Hypermedia systems (EAH) should adapt how the information is presented, not just how the learner navigates through the information. Our research explores the connection between presentation and user engagement. Results of this research will allow us to adapt presentation in EAH to maximize learner engagement.

2. ACUT

While most learning occurs outside the bounds of formal education, most research in AH has been based on metaphors from formal education, such as lecture, textbook, homework, or one-on-one tutoring. In contrast, our laboratory focuses on spontaneous small group learning, a metaphor drawn from informal education. Our design metaphor constrains us to modeling learners unobtrusively, as informal education requires sharing control over the timing and shape of learning with the learner [1].

We will base our experiments on the most current version of an existing informal EAH, the Adaptive Collaborative UNIX Tutorial (ACUT) [1]. ACUT unobtrusively collects user subsymbolic behavioral data. It is unique in that it does not require any modification of the user's browser, which makes its method of data collection suitable for use outside of a laboratory setting [2].

3. Modeling Engagement

We are developing an unobtrusive cognitive learner model derived from personality traits of the learner and the structural and graphical features, rather than

semantic content, of hypermedia documents. The semantic content that a user finds interesting may change over time. The style of interface that a learner finds engaging is a product of cognitive style, which is relatively stable over time [3].

Since the presentation style is what makes a document engaging [4], our central hypothesis is that adapting the structural and graphical content of documents will improve learner engagement with an informal EAH.

Measuring engagement is difficult, as it requires modeling a user's mental state. We can measure engagement only indirectly, e.g. via preference for particular types of resources. An informal EAH may have multiple resources with the same, or similar, semantic content. In this circumstance, a high relative rate of revisitation of a particular document would strongly indicate a preference for its non-semantic features. We propose the use of user preference as measured by revisitation as an indirect, unobtrusive behavioral measure of engagement. For our study, we will use the formula defined by Flesca [5] for determining preference via revisitation.

The documents in the system will be clustered based on structural and graphical content. The features for clustering will be designed in a phase of exploratory data analysis using self-organizing maps [1]. We will determine the users' preference for each cluster of documents. ACUT will adapt presentation based on these online learning styles.

4. Conclusion

We have created a design methodology for investigating the problem of providing an unobtrusive measure of engagement for use in EAH. Because our method is unobtrusive it may be implemented in any real-world system without diminishing its usability. Our next step is to empirically validate this methodology with results from its implementation on our ACUT platform and apply our results to adapt presentation in ACUT.

Notes

[1] ACUT can be accessed online at http://acut.csueastbay.edu/

References

[1] H.J. Holz, K. Hofmann, C. Reed, Unobtrusive User Modeling for Adaptive Hypermedia, *Intl. Journal of Pattern Recognition and Artificial Intelligence*, in press.
[2] K. Hofmann, C. Reed, and H. Holz. Unobtrusive Data Collection for Web-based Social Navigation. In *Wk. SNC-BAT, AH2006*
[3] A. Dufresne and S. Turcotte, "Cognitive style and its implications for navigation strategies," in *AIED'97*, no. 8. Amsterdam: IOS, 1997.
[4] P. Chapman, S. Selvarajah, and J. Webster, "Engagement in multimedia training systems," *hicss*, vol. 01, p. 1084, 1999.
[5] S. Flesca, S. Greco, A. Tagarelli, and E. Zumpano, "Mining user preferences, page content and usage to personalize website navigation," *World Wide Web*, vol. 8, no. 3, pp. 317–345, 2005.

R. Luckin et al. (Eds.)
IOS Press, 2007

Applying Learning Factors Analysis to Build Stereotypic Student Models

Anna N. Rafferty[a] and Michael Yudelson[b]
[a] *Stanford University, rafferty@stanford.edu*
[b] *University of Pittsburg, mvy3@pitt.edu*

In this research, we examine the issue of increasing the abilities of intelligent tutoring systems to consider individual student learning patterns as well as how students learning patterns in these systems may resemble one another. The need to examine how students' learning may systematically differ is motivated by the fact that some intelligent tutoring systems are hampered by the large increase in complexity that results from adding many student specific modifications [1]. Additionally, understanding systematic patterns in learning may improve our understanding of why students' success rates in learning new skills differ and allow intelligent tutoring systems to take advantage of these differences.

Our work depends on Learning Factors Analysis (LFA) [1] to model student knowledge. LFA is an extension of the power law of learning [1], which represents the exponential decrease in error rate that occurs with increasing numbers of opportunities to practice a skill. While this power law is defined only over one skill and one student, LFA models multiple students and multiple skills by adding student intercepts, skill intercepts, and skill learning rates. This increases the space of models to allow for students with different initial skill levels and skills with different initial difficulties and learning rates. Notably, however, this model does not allow for variations in the rate at which each student learns; Cen et al. [1] comment that this step was justified in their application because they were attempting to improve the model as a whole rather than improve its ability to model individual students. We seek to extend the potential applications for this technique by allowing different groups of students to have different learning rates.

To examine the feasibility of this technique, we used data from a geometry cognitive tutor to group students based on learning and knowledge characteristics; we then fit models for each group of students to examine learning patterns for each group. Students were categorized into groups by calculating a measure of learning gain and initial knowledge for each student and categorizing students as showing high or low learning gain and high or low initial knowledge. Using these two types of splits, students were broken into four groups of approximately equal size. Students' average success rates on their first encounters with each skill were only weakly correlated with their learning gain (r^2=0.17), showing the splits represent two different characteristics.

While some changes in model fit were found when splitting students based solely on learning gain or initial knowledge, more profound differences were found when LFA was applied to groups based on both of these measures. First, when comparing learning rates, significant differences were found between groups. This is notable

because the original model could not express these differences in learning rates, providing an empirical challenge to Cen et al.'s [1] assumption that learning rates can be held constant across students. Particularly pronounced effects were found when comparing students with low initial knowledge who had high versus low learning.

Another compelling result was found by adjusting the skills in the cognitive model. We compacted the skill model by grouping the fine-grained skills into coarse-grained units and refit the models. As suggested empirically by Koedinger and Mathan [2] and shown theoretically by the learning curve equations, a compact model allows for faster mastery of skills if learning rates do not decrease sufficiently to compensate for the greater number of practice attempts for each skill. We found that the compact model fit the students with low initial knowledge and high learning gain best; this suggests these students are learning a domain model in which each skill attempt is equivalent to multiple attempts on different skills for the other groups.

Our research demonstrates that it is not sufficient to use a uniform learning rate to model students; students clearly differ not only in their initial knowledge but also in the number of practice attempts they require to learn new skills. Future research should examine whether the differences in learning rates and skill granularities between students can be estimated for new subdomains based on performance in a previous, related subdomain to suggest ways of incorporating this work into educational tutors.

Despite the need for further investigation, this study provides valuable information about what assumptions can be made about learners. By suggesting a hybrid approach to considering individual student characteristics, we hope to encourage those creating cognitive models to incorporate traits that vary among students. Additionally, we seek to increase interest in understanding precisely what students are learning and whether student domain models are consistent across students and match tutor domain models.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. SLC-0354453. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. We would like to acknowledge the Pittsburgh Science of Learning Center Summer School for providing the training and opportunity to conduct much of this research, and to thank Daniel Schwartz, Joseph Beck, and Kenneth Koedinger for their insightful comments on this research as it progressed, and Hao Cen for his insight and assistance with LFA.

References

[1] Cen, H., Koedinger, K. R., & Junker, B. (to appear). Learning Factors Analysis – A general method of cognitive model evaluation and improvement. In Proceedings of Intelligent Tutoring Systems, 2006.

[2] Koedinger, K. R., & Mathan, S. (2004). Distinguishing qualitatively different kinds of learning using log files and learning curves. In the Working Notes of the ITS2004 Workshop on Analyzing Student-Tutor Interaction Logs to Improve Educational Outcomes.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

A Student Model Based on Item Response Theory for TuTalk, a Tutorial Dialogue Agent

Michael RINGENBERG
mringenb@pitt.edu; LRDC #743, 3939 O'Hara St., University of Pittsburgh, Pittsburgh PA 15260

Intelligent tutoring systems are effective tools in educational settings and are also valuable as platforms for conducting educational research [1]. They can be effective at promoting learning, but still lack the power of one-to-one human tutoring [2]. One way to help bridge the gap between human and computer tutors may be introducing dialogue, which has been made possible with advances in natural language understanding. This principle is the foundation of TuTalk, a system for authoring and running dialogue agents, particularly as part of intelligent tutoring systems [3].

A domain expert can use TuTalk to build knowledge construction dialogues (KCDs) to help guide a student to construct missing knowledge or repair misapprehensions in their mental model. A basic KCD usually starts off with a bit of exposition, usually to introduce a problem or knowledge component, and a probing question. If the student answers the question correctly, the system will move on to the next KCD. If the question was answered incorrectly, then the student enters a subdialogue whose goal is to remediate the incorrect knowledge component. If the student's answer is not recognized, the student can be presented with a dialogue that covers the knowledge component in a general way.

Multiple KCDs can be authored to cover the same concept. In this situation, TuTalk chooses which of these KCDs to present based on its student model and the difficulty level of the KCD set by the author. TuTalk currently implements a form of Wood's contingent tutoring [4]. It will check the difficulty level of the previous question and if the student provided a correct answer, then it uses the KCD that has the next hardest difficulty level, otherwise it uses the next easiest level.

However, it should be possible and beneficial to examine all of the student's responses in order to estimate competence. Recent work in data-centric approaches to student modeling supports using Item Response Theory (IRT) in a tutoring system for geometry [5]. IRT is based on the assumption that responses to assessment items are a function of both the competence of the student and the difficulty of the item and can be used to estimate how difficult a given question would be for a given student. This could allow the system to choose optimal KCDs for each student to maximize learning.

The structure of TuTalk dialogues allow the application of an IRT model. While KCDs are not a classic series of multiple choice questions, it can be easily seen as such because TuTalk is matching the student's answers to a finite list of answers and the answers are dichotomously scored (marked as either correct or incorrect). As for using the IRT model for KCD selection, one possible rule could be to pick the most difficult KCD in which the student still has a good chance (at least 50%) of getting the first question correct. Multiple rules will be investigated once data are collected.

There will be two main methods for evaluating the usage of an IRT student model in TuTalk. The first method will be to see if a trained IRT student model is a good predictor of performance. The second method will compare which choices of KCDs different student models would make.

There are several avenues of further research that can be investigated if the initial findings are promising. One would be to investigate different IRT models. There are IRT models that can account for repeated questions [6] or for learning [7]. Additionally, a finer grained IRT model could be used in which each of the knowledge components are given individual assessments of competence as opposed to a single assessment for the entire scenario.

The IRT student model could also be incorporated into the authoring process in various useful ways. For example, the difficulty of the questions as determined by the IRT model could be provided to the author who could then determine if there are sufficient KCDs to cover a variety of competences or to rework questions that are too difficult or too easy.

References

[1] Aleven, V., Sewal, J., McLaren, B.M., Koedinger, K.R.: Rapid authoring of intelligent tutors for real-world and experimental use. In Kinshuk, Koper, R., Kommers, P., Kirschner, P., Sampson, D.G., Didderen, W., eds.: 6th IEEE International Conference on Advanced Learning Technologies (ICALT 2006), Los Alamitos, CA, IEEE Computer Society (2006) 847–851

[2] Bloom, B.S.: The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. Educational Researcher **13** (1984) 4–16

[3] Jordan, P., Ringenberg, M., Hall, B.: Rapidly developing dialogue systems that support learning studies (2006)

[4] Wood, H., Wood, D.: Help seeking, learning and contingent tutoring. Computers and Education **33**(2-3) (1999) 153–169

[5] Johns, J., Mahadevan, S., Woolf, B.: Estimating student proficiency using an item response theory model. In: Intelligent Tutoring Systems, Proceedings. Volume 4053 of Lecture Notes in Computer Science. Springer-Verlag Berlin, Berlin (2006) 473–480

[6] Andersen, E.B.: Estimating latent correlations between repeated testings. Psychometrika **V50**(1) (1985) 3–16

[7] Susan, E.E.: A multidimensional latent trait model for measuring learning and change. Psychometrika **V56**(3) (1991) 495–515

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Automating the Generation of Student Models for Intelligent Tutoring Systems

John Stamper
Department of Computer Science
University of North Carolina at Charlotte
9201 University City Blvd., Charlotte, NC 28223
jcstampe@uncc.edu

Keywords. Intelligent Tutoring, Adaptive Learning, Educational Data-Mining

Introduction

Intelligent Tutoring Systems (ITSs) that adapt to an individual student have been shown to be extremely effective, showing significant improvement in achievement over non-adaptive instruction[3]. However, the most successful of these systems require the construction of complex cognitive models that are applicable only to a specific tutorial in a specific field, requiring the time of experts to create and test these models on students. In order to achieve the benefits that ITSs provide, we must find a way to simplify their creation. Therefore, we are creating a framework to automate the generation of ITS student models. The goal is to provide a simple way to allow developers of computer-based training (CBT) to add adaptive capabilities with minimal work while still maintaining the effectiveness of a true ITS.

1. Background and System Design

Student and domain knowledge models are central tools for adaptation, but creating these models is the bottleneck in time and effort in creating ITSs. We believe that using educational data mining techniques to build the cognitive model from existing CBTs can be less time consuming and will also produce effective student models. Our system is modeled on ACT-R theory[1], using a cognitive architecture that uses production rules to model student problem solving processes. Model Tracing(MT) and Knowledge Tracing(KT) are the two main processes used to adapt ITS behavior to the student. MT tracks a student's progress through a particular problem, matching their steps to production rules, and providing feedback if they proceed down an unsuccessful path. Production rules can be "good" rules that are correctly applied, or "bad" rules, which are generally student misconceptions. We used data from an existing CBT to test the idea of using Markov Decision Processes to learn production rules for MT[4]. We compared the resulting production rule sets to those generated by experts. Our methods were able to extract all expert-predicted approaches, both correct and incorrect, while also revealing an extensive set of incorrect approaches that were not predicted by experts. KT tracks a student's overall learning of concepts within the system. Typically,

ITS designers create a skill matrix to relate problems to their respective knowledge components. These skill matrices can also be learned using educational data-mining techniques such as the q-matrix method[2], which is a data-mining algorithm that extracts a q-matrix from student assessment data and will discover “concepts” that influence student behavior. This method has been successfully applied to learning the concepts underlying an online physics tutor, and these concepts compare well to the expert-derived Facets that were used to create the tutor[2]. We believe that the method can be used to automatically update the KT model.

Our system will be comprised of a Knowledge Tracing Module (KTM) that will assess and direct student progress, and a Model Tracing Module (MTM) to provide problem-specific feedback. These modules will be added to an existing CBT called Deep Thought (DT) that is used to teach propositional logic. The KTM will make decisions based on a student state relative to the overall skill matrix for the problems being solved. The skill matrix will be initially derived from historical data, and will be continually updated as new students use the system. The MTM will match student performance to a production rule system, initially derived using the MDP methods from our previous work, and will also be continually updated to include new rules to describe student behavior. Probabilities associated with a particular path through the production rule system are updated by every step that a student makes. The system then predicts a student’s next action based on both the current rules associated with the problem being worked on and the student’s current classification, which is derived from the KTM. When probabilities are reliable for a student, feedback can be given when appropriate.

2. Conclusions and Future Work

We have successfully demonstrated that we can automatically generate the parts of our student model, and we have designed the combined model for integration into a specific CBT system. We can derive production rules for logic proofs using MDPs, and believe we will be able to generate appropriate feedback based on these rules. We can derive a concept model using the q-matrix method, and use this model to assess and predict student knowledge. These basic components provide the foundations for giving a CBT system full adaptive capabilities of a traditional ITS. We plan to validate the predictions the system generates about student progress, and also create a feedback generation system. Later, we will compare performance using our new system to using DT alone to determine if significant improvements are made.

References

[1] Anderson, J. R., Corbett, A. T., Koedinger, K. R., & Pelletier, R. 1995. Cognitive Tutors: Lessons Learned. The Journal of the Learning Sciences, 4(2), 167-207.

[2] Barnes, T., Stamper, J., Madhyastha, T. 2006. Comparative Analysis of Concept Derivation Using the Q-matrix Method and Facets. In Educational Data Mining: Papers from the 2005 AAAI Workshop, ed. Joseph E. Beck, 21-30. Technical Report WS-06-05. AAAI, Menlo Park, California

[3] Murray, Tom. 1999. Authoring intelligent tutoring systems: An analysis of the state of the art. International Journal of Artificial Intelligence in Education, 10: 98-129.

[4] Stamper, J. (2006) Automating the Generation of Production Rules for Intelligent Tutoring Systems. Proceedings of the 9th International Conference on Interactive Computer Aided Learning, 206-212,Villach, Austria.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Supporting Learning in Games with an Open Learner Model

David Storey
School of Information Technology, University of Sydney, Australia

Keywords: Computer, Games, Performance, Measurement, Learning

Introduction

Every day millions of people around the world play computer games for fun and entertainment. The games they play are varied yet share one thing in common – in order to play them at an acceptable level the user first had to learn how to play, and the game provides a very motivating environment within which to do so.

It is this property of learning in a fun environment that has attracted the interest of the educational community. People have examined the potential that games hold, used games as a tool to teach certain concepts, as a training and testing environment for artificial intelligence (some game-related, some not) and more recently have taken to online games/environments (commonly known as virtual worlds) and created educational spaces within them. In industry an entire gaming genre consisting of educational games exists and met with some success.

All of these share one similarity – they use the game as a tool, attempting to fit their educational content into the gaming environment instead of using the existing gaming environment to achieve educational outcomes. Unfortunately mainstream games are rarely built for education or learning outside of its own gameplay and there currently exists no standard way to obtain, process and use game data that can theoretically be applied to any game to support learning.

We set out to solve this problem by developing a process which can be applied to a game in order to obtain the necessary data and measure a player's performance. This lead to a functioning prototype based on "Quake 3: Arena" and the aim of this prototype was to objectively measure an individual's performance within the game. Quake 3 was chosen because of its simplicity, wealth of community knowledge, and easy access to game data through an official modification kit.

Our previous work involved testing with built-in computer players to verify the criteria produced reasonably accurate results for them. After the success of those tests we have now tried it with a human Quake 3 expert in order to continue testing accuracy and also to test the feedback mechanism and identify potential problems with our method that would restrict its applicability to other games of either the same genre or other genres.

1. Prototype System and its Performance

Our prototype system collects event data from Quake 3 through a custom modification. This modification changes none of the game mechanics and does not impact gameplay at all. Once data is collected from a round of play it is saved and can be analysed with a stand-alone application that measures 5 criteria. These are the player's accuracy, pause frequency, choice of weapons in battle, avoidance of enemy fire and number of kills. It generates both summary and detailed results that they player can examine.

These criteria were chosen from a selection that were generated by analysing the information available from the online Quake 3 community coupled with the authors personal experience from several years of playing Quake and other shooter-style games.

To test our current system we obtained the services of a Quake 3 expert, had them play several rounds against computer controlled opponents. We then discussed their performance, introduced them to our measurement system and examined how the perceptions of performance matched up with the numerical results, before discussing how useful they felt the display was for self-analysis and as a tool for self-improvement.

The results of that experiment showed that there were some differences between the experts expected and actual performance, partly explained by technical issues, and also showed that the display did provide a good summary but fell short of providing enough detail to help them identify specific areas of improvement.

2. An improved measurement system

Our system is functional yet far from perfect. One particular problem is that generalising it to other games is not easy – not all games have a body of community knowledge (particularly new games) and personal knowledge has limitations.

Thus we have developed an improved system which uses a detailed analysis of the game to help identify performance criteria. It starts by examining the games objectives and then examining what tasks the player must carry out to achieve those objectives. Those tasks are then examined to see what other tasks are needed to complete them successfully. This repeats until a basic action of the game is reached, represented by an event. In addition the situations players get into whilst carrying out their tasks are considered as are actions which would prevent opposing players carrying out their own tasks, thus slowing their progress.

Using the new system we can rationalise our current criteria as follows. The overall objective in the standard game mode is to reach a certain number of kills before your opponents do. You achieve this by killing players (Kill Count criteria), which requires you hurt them, which requires you to shoot at and hit them (Accuracy). To shoot at the players you need to find them, which requires constant movement (Continuous movement). Weapon selection can be generated by examining the situation a player gets into when they start firing at and fighting with someone - a bigger weapon has a higher probability of winning the fight. To prevent your opponent killing you their shots need to be avoided, which leads to Dodging.

The new approach also stipulates levels of detail for criteria, emphasising a layered approach to development with each layer providing summary details instead of a direct leap from raw data to the final result.

In total the improvements will lead to a system that can be adapted easier to different games and generates results that provide more useful feedback to players.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Developing a General Model for Supporting Explanations

Amali WEERASINGHE
Intelligent Computer Tutoring Group
University of Canterbury, Christchurch, New Zealand
acw51@cosc.canterbury.ac.nz

Students acquiring shallow knowledge is a problem in many learning environments including intelligent tutoring systems (ITS). Such students learn enough to pass exams but have difficulties applying their knowledge in new and unfamiliar situations. Empirical studies indicate that some shallow learners attempt to *game the tutoring system* in order to complete the problems. Therefore, researchers have long been investigating strategies that can foster deep learning. Self-explanation (SE) is one such strategy that has been effective in facilitating deep learning. Several ITSs have been enhanced with self-explanation support in domains such as physics, mathematics, database design and data normalization. All these instructional tasks except database design are well-defined, as problem solving is well structured, and therefore self-explanation expected can be clearly defined. Database design is an ill-defined task: the final result is defined in abstract terms, but there is no algorithm to find it.

Our long-term goal is to develop a general model of explanations which will provide adaptive support to learners across domains. The main objective of this model is to assist students to develop their self-explanation skills while being prompted to explain their mistakes to the tutor. Since we previously implemented explanation support for the database design tutor, the initial work on this project started with the same tutor. As the first step, we conducted an observational study focusing on how students interacted with EER-Tutor, while getting additional help from a human tutor through a chat interface. A model to facilitate explanations was then developed.

The explanation model is used to decide when to prompt for explanations, what to explain and how to obtain explanations from learners. The model consists of three parts: error hierarchy, self-explanation dialogues and rules for adapting them. Error hierarchy and the dialogues are used to determine timing and content of the explanations respectively. Learners are given the opportunity to explain by selecting the correct explanation from a list provided by the tutor.

Error Hierarchy: The domain model of constraint-based tutors is represented as a set of constraints. Violations of constraints indicate mistakes in students' solutions. In previous work, we developed a hierarchy of errors students make in the Entity-Relationship (ER) domain, which categorizes errors as being syntactic or semantic in nature. Syntax errors are simple, each requiring only one feedback message to be given to the student; for that reason, every syntactic error corresponds to a particular constraint being violated. There are 12 such constraints. The hierarchy for semantic errors is deeper, with error types further divided into sub-errors. These nodes are ordered from basic domain principles to more complicated ones. Violated constraints

for each type of error are represented as leaves of the hierarchy. As the ER error hierarchy was developed for a particular domain, we were interested whether it can be reused in other domains. With that goal, we tried to fit the errors for ER-to-relational-mapping, data normalization and fraction addition. As all these tasks use deterministic algorithms, the tasks are well-defined.

The error hierarchy needed to be refined to accommodate different domains. The highest level of the refined hierarchy has two nodes: *Basic syntax errors* and *Errors dealing with the main problem solving activity*. The second node is now further categorized into five nodes: (i) *Using an incorrect construct type* (ii) *Extra constructs* (iii) *Missing constructs* (iv) *Associations* and (v) *Failure to complete related changes*. The subsequent levels deal with domain-specific concepts. The common feature in all these tasks is that the syntactic and semantic accuracy of a solution can be completely evaluated by the components of the solution and its associations. However, there are exceptions. For instance, in reading and comprehension, where learners are asked to answer questions based on a paragraph, the accuracy of an answer cannot be evaluated by checking for correct words according to the grammar rules. We also need to understand the implicit semantic meaning of the sentence. Our error hierarchy is not useful in such cases. In summary, we have been able to use this hierarchy in four different types of tasks: thus we believe it would be sufficiently general to be used for different types of instructional tasks only when the solution can be completely evaluated by the components of the solution and its associations.

Tutorial Dialogues: Explanation is facilitated through tutorial dialogues. For each error type (i.e. each leaf node in the hierarchy), we designed a dialogue consisting of four stages. In the first stage, the dialogue informs the student about the concept that s/he is having difficulty with. For example, *You seem to be having some difficulty with regular entities. Let's look at regular entities in detail. Can you tell me the general rule to decide whether something is a regular entity?* (Prompt1). The purpose of the second stage is to assist the student in understanding why the performed action is incorrect. The third stage prompts the student to specify how to correct the mistake. In the fourth stage, the student can review the domain concept learned. Although the prompts are domain-specific, the structure of the dialogues is domain-independent. We are currently investigating the applicability of the dialogue structure to various domains

Rules for adapting dialogues: These rules individualize dialogues by deciding on the entry point and/or the timing of the dialogue. Currently there are eight rules and they are based on a previous study conducted using EER-Tutor. For example, rule 4, dealing with customizing the entry point to the dialogue, is initiated when the same error is made in the last *n* attempts. In that case, a dialogue corresponding to the mistake will be initiated, but the dialogue would start from the problem-independent question (Prompt 1 above). If the error is made less than *n* attempts, then the dialogue starts from the error within the current context. As these rules do not depend on domain specific details to individualise dialogues, the rules can be used across domains.

The next step is to incorporate the model into two existing tutoring systems EER-Tutor and NORMIT that provide assistance to learn database design and data normalization respectively. These enhanced systems will later be evaluated in authentic classroom environments.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Incorporating Reflection Prompts into Learning-by-Teaching Environment to Foster Reflective Student Teacher

Longkai WU
National Institute of Education, Nanyang Technological University
1 Nanyang Walk, 637616 Singapore
wulk@mail.nie.edu.sg

Introduction

Past research has emphasized on the importance of reflection in teaching and learning. John Dewey (1910, 1933) said "Experience plus reflection equals growth" and advocated the use of reflection to enable teachers to learn from their experience. Schön (1983) suggested that the capacity to reflect on action so as to engage in a process of continuous learning was one of the defining characteristics of professional practice. Raelin (2001) proposed that reflection can illuminate "what has been experienced" and provide "a basis for future action". Davis (1998) used the term "reflection" to refer to both metacognition and sense-making, i.e., reflection can focus on one's own thinking or goals, or on content itself.

Biswas, Schwarz, Leelawong and et al. (2001,2003) proposed and developed learning by teaching systems, called teachable agents, based on the adage that teaching others is a powerful way to learn. The teachable agent environment enables a middle-school student to play the role of a teacher in facilitating learning. The student teaches a computerized agent with explicit instruction and the agent will solve problems independently and display the results to the student teacher. It is claimed that this procedure can motivate the student to reflect on what they have taught and gain deeper understanding of the subject material. However, we argue that the middle schools students still lack the ability to autonomously reflect on their goals, contents and thoughts in their first exposure to doing some teaching practice. They need further support from the learning by teaching environment to participate in reflections for better learning.

In this paper, we introduce reflection prompts into a learning-by-teaching environment to foster and develop reflective student teachers who are encouraged to reflect on and self-explain their teaching experiences. A reflection prompting schema is designed that consider three different aspects i) the timing of providing reflection prompts ; ii) the target levels of reflection; and iii) the types of reflection prompts. To implement the designed schema, we propose a new agent architecture that is capable of monitoring the students' interaction with the learning-by-teaching environment and produce appropriate reflection prompts tailored to students' levels of learning. Pilot study on the benefits of introducing reflection prompts and the further development of the Learning-by-Teaching Environment with reflection support is described

1. Pilot Study and Results

To assess potential effectiveness and implications for designing reflection prompts in learning by teaching environment, we conducted a pilot study on 10 female students from a local secondary girls' school. The learning by teaching environment we adopted was a revised basic version of Betty's Brain which was originally developed in Teachable Agent Group at Vanderbilt University to teach middle school students about the concepts of interdependence and balance in river ecosystem (Biswas et al. 2001). We built a new domain: basic concepts of supply and demand in economics.

The students were divided into two groups, the *teach* group and the *reflection* group. The teach group members taught Betty by creating the knowledge structures, generating questions, and analyzing Betty's responses to questions. The reflection group students, on the other hand, were asked to write down reflection notes to their ongoing teaching practices triggering by the four categories of reflecton prompts from pre-prepared sheets and human tutors in the classroom.

Analysis of the scope of students' concept maps shows that the average number of valid concepts used by the reflection group (M=6.6, SD=1.52) was greater than the teach group (M=5.8, SD=1.64), F (1.8=0.64, p=0.45. The average number of valid links used by the reflection group (M = 13.8, SD = 2.68) was greater than the teach group (M = 8.6, SD = 2.19), F (1, 8) = 11.27, p = 0.01.

2. Discussion and Future Work

Our study with reflection prompts in a learning-by-teaching environment demonstrates their potential in promoting involvement in learning among students. Students require additional support, like reflection prompts, to learn and practise the general metacognitive skills, task-specific skills and domain knowledge system for implementing learning. Feedback from the secondary school class and their teachers indicate that the idea of learning by teaching was successful in motivating students and getting them to spend more time in learning on complex and unfamiliar domains. More extensive studies will be conducted with a focus on incorporating the reflection prompts into the system and to evaluate their effects on students' learning in the environment.

References

[1] Dewey, J (1910, 1933). How we think. Boston, D.C. Heath & Co.
[2] Schon, D. (1983). The reflective practitioner: How professionals think in action. NewYork: Basic Books.
[3] Raelin, J.A. (2001). Public reflection as the basis of learning, Management Learning, 32(1), 11-30.
[4] Davis, E.A. (1998). Scaffolding students' reflection for science learning. Unpublished doctoral dissertation, University of California, Berkeley, CA.
[5] Biswas, G., Schwartz, D., Bransford, J. (2001). Technology Support for Complex Problem Solving: Smart Machines in Education, 71-98. Menlo Park, CA: AAAI Press.
[6] Leelawong, K., et al. (2003). "Teachable Agents: Learning by Teaching Environments for Science Domains," Proc. 15th Innovative Applications of Artificial Intelligence Conf., Acapulco, Mexico, 109-116.
[7] Gama, C. (2004). Metacognition in Interactive Learning Environments: The Reflection Assistant Model. Int C on Intelligent Tutoring Systems, 668-77.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Are We Asking the Right Questions? Understanding Which Tasks Lead to the Robust Learning of the English Article System

Ruth WYLIE
Human-Computer Interaction Institute, Carnegie Mellon University
5000 Forbes Ave., Pittsburgh, PA 15213, USA
rwylie@cs.cmu.edu

Article usage has been cited as one of the most frequent causes of error and one of the most difficult grammar points for English language learners (ELL) to learn [1]. It is not uncommon for students to complete worksheets successfully on the topic but continue to make errors during authentic production. This paper presents work that attempts to bridge the link between practice and production using computer-based tutoring systems. Motivated by both classroom needs and learning science questions, we built two tutoring systems using the Cognitive Tutoring Authoring Tools (CTAT) [4] to help students learn to make distinctions regarding English articles (*a*, *an*, *the*, and null). The first is a menu-based tutor that mimics the fill-in-the-blank exercises found in many textbooks. Using this interface, students select an article from each drop-down menu in order to complete the paragraph. Students do not need to identify where errors exist; they simply choose a response for each box (Figure 1). The second tutoring system helps students learn both error detection and error correction skills via a controlled-editing tutor. Unlike in the menu-based task where students know exactly where changes need to be made, in the controlled-editing tutor, students must first identify where the error exists and then correct it (Figure 2). Both tutors provide immediate feedback and detailed hints to help students complete the task.

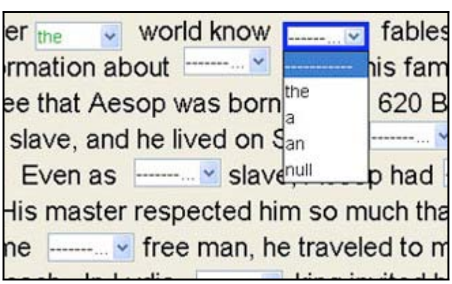


Fig. 1. Menu-based tutor

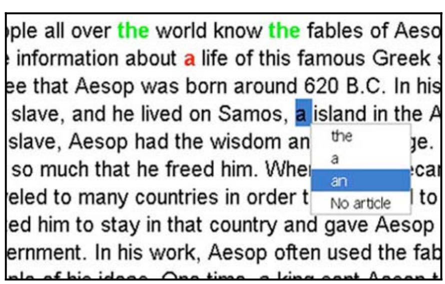


Fig. 2. Controlled-editing tutor

A study was conducted in order to examine differences between the two task types as well as gather verbal protocols in order to understand which rules and heuristics students employ when solving these problems. If students performed equally well (or poorly) using the different interfaces, one would not expect to see differences in learning. In answering these questions, we performed a think-aloud study in which we invited advanced ELL participants (n=6) into the lab and asked them to complete tasks using the two systems. Students were told beforehand that the only errors present within the paragraphs were article errors.

The problem paragraphs came from intermediate and advanced-level ESL textbooks (e.g. [2], [3]). The first problem was shorter in length and used simpler vocabulary than the second problem did. Students were randomly assigned to one of two groups: those in the first group completed the intermediate level problem using the controlled-editing interface and the advanced level problem with the menu interface. Students in the second group did the opposite.

The data reveal a strong performance difference between the two tasks. For both problems, students using the menu-based tutor were more accurate, as measured by the percent of necessary changes that were correctly made. While the difference between the two interfaces was not significantly different for the easier, intermediate-level problem ($t=1.45$, $p = 0.13$), the performance results were significantly different for the advanced problem ($t=2.13$, $p = 0.05$), suggesting that when students are presented with level-appropriate texts, detecting errors is a formidable challenge.

Learning the English article system is a challenging but necessary process for students writing academic and professional texts. Thus, it is important to understand which tasks best support knowledge acquisition, retention, and transfer. This work marks the first steps of this process: understanding the educational and learning science motivations, developing the tutoring systems, and conducting initial student studies. While the results show significant performance differences between the menu-based and controlled-editing tasks, the question, and focus of current work, is whether this added difficulty will lead to greater learning gains. Current plans also include moving the tutors out of the lab and into the classroom in order to gather authentic data from a more diverse student sample.

Acknowledgements. This work was conducted within the Pittsburgh Science of Learning Center under NSF Award #0354420. Special thanks to Teruko Mitamura, Ken Koedinger, Jim Rankin, and three anonymous reviewers for their suggestions and feedback.

References

[1] Celce-Murcia, Marianne, and Larsen-Freeman, Diane. (1983). *The Grammar Book: An ESL/EFL Teacher's Course*. Rowley, Massachusetts: Newbury House Publishers.

[2] Fuchs, M., Bonner, M., & Westheimer M. (2006). *Focus on Grammar 3: An Integrated Skills Approach, Third Edition.* White Plains, NY: Pearson Education, Inc.

[3] Huckin, T. & Olsen L. (1983). *English for Science and Technology: A Handbook for Nonnative Speakrs*. McGraw-Hill, Inc.

[4] Koedinger, K. R., Aleven, V., Heffernan. T., McLaren, B. & Hockenberry, M. (2004). Opening the Door to Non-Programmers: Authoring Intelligent Tutor Behavior by Demonstration. In the Proceedings of 7th Annual Intelligent Tutoring Systems Conference. Maceio, Brazil.

Workshop Summaries

This page intentionally left blank

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

AIED Applications in Ill-Defined Domains

Vincent ALEVEN [a], Kevin ASHLEY [b], Collin LYNCH [c] and Niels PINKWART [d]
[a] *HCII Carnegie Mellon University, aleven@cs.cmu.edu*
[b] *School of Law, University of Pittsburgh, ashley@pitt.edu*
[c] *ISP University of Pittsburgh, collinl@cs.pitt.edu*
[d] *Clausthal University of Technology, niels.pinkwart@tu-clausthal.de*

Keywords. AIED Ill-Defined

AI Supported Educational systems have made great strides in recent years both as research tools and teaching applications. Most of the AIED research and development to this point have been in *well-defined* domains such as Physics, and Mathematics. Such domains are characterized by a well-accepted theory or model that makes it possible unambiguously to classify problems as correct or incorrect.

Not all domains of teaching and inquiry are well-defined, indeed most are not. Domains such as law, argumentation, history, art, medicine, and design are *ill-defined* [4]. Often even well-defined domains are increasingly ill-defined at the edges where new knowledge is being discovered. Ill-defined domains lack widely supported models and theories that can be operationalized for model-based tutoring or expert systems. Thus they present a number of unique challenges for AIED researchers.

While some educational tools have been developed for ill-defined domains such as Music [3] and Law [2] little effort has yet been made to weld these disparate efforts into a cohesive research community. This workshop is a continuation of our very successful workshop in 2007 [1] which began a discussion on the use of novel and well-tested tutoring techniques in ill-defined domains as well as the assessment of such approaches. We fully expect this workshop to go further in helping to build a solid research community in this ever expanding area.

References

[1] Aleven, V., Ashley, K., Lynch, C., & Pinkwart, N. (Eds.) (2006). “Proceedings of the Workshop on Intelligent Tutoring Systems for Ill-Defined Domains” at the *8th International Conference on Intelligent Tutoring Systems.* Jhongli (Taiwan), National Central University.

[2] Ashley, K.D., Desai, R. and Levine, J.M. (2002) “Teaching Case-Based Argumentation Concepts using Dialectic Arguments vs. Didactic Explanations”. In *Proceedings, Intelligent Tutoring Systems Conference, ITS ’ 02* (S.A. Cerri, G. Gouardéres, F. Paraguaçu, ed.) Lecture Notes in Computer Science. Pp. 585-595. Springer-Verlag: Berlin.

[3] Holland, S. (1999). Artificial intelligence in music education: A critical review. *Readings in Music and AI; Contemporary Music Studies,* **20**.

[4] Lynch, C., Ashley, K., Aleven, V., & Pinkwart, N. (2006). “Defining Ill-Defined Domains; A literature survey.” In V. Aleven, K. Ashley, C. Lynch, & N. Pinkwart (Eds.), *Proceedings of the Workshop on Intelligent Tutoring Systems for Ill-Defined Domains* at the 8th International Conference on Intelligent Tutoring Systems (p. 1-10). Jhongli (Taiwan), National Central University.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Assessment of Group and Individual Learning through Intelligent Visualization (AGILeViz)

Eric HAMILTON [a], Amy BAYLOR [b], Mark CHAVEZ [c], Ron COLE [d], Chris DIGIANO [e], Andy HURFORD [a], Michael JACOBSON [f], Manu KAPUR [f], Leo LEE [g], Richard LESH [h], Manuel LIMA [i] and Philip VAHEY [e]

[a] *Center for Research on Learning and Teaching, Institute for Information Technology Application, US Air Force Academy;* [b] *Florida State University;* [c] *Nanyang Technological University School of Art, Design, and Media;* [d] *Center for Spoken Language Research, University of Colorado at Boulder;* [e] *SRI International;* [f] *Learning Sciences Laboratory (LSL) at the National Institute of Education (NIE) of Singapore;* [g] *Sparkpoint Consulting;* [h] *Indiana University;* [i] *Parsons School of Design*

Purpose of the Workshop

The workshop will discuss how to provide educators with sophisticated, computer-generated visualizations of learning by groups and individuals. The workshop is motivated by the convergence of several factors that bear directly on the AIED07 vision to build technology-rich learning contexts that work:

- Developers of intelligent tools for learning, and educational innovators more broadly, use technology to understand and propel learning in ways that dwarf the limited paradigms of assessment tools dominating educational practice.
- Current assessment tools not only are highly constrained in the constructs that they measure, but the representational systems used to communicate student progress through such assessments are limited. Representational systems (e.g., single value test reports, linear summaries of mastery estimates, etc.) are inadequate for representing progress in the kinds of learning and problem-solving behavior that will be increasingly important to assure in the future.
- Interactive information visualization techniques for large quantities of heterogeneous data have now reached a state of maturity that make them ripe for application to the domain of data-drive decision making in education.
- Advances in graphics capabilities of personal computers mean that responsive 3D interfaces are now affordable to many educators.

Our goal is to nurture the nascent area of applying complex graphical systems for assessment of complex individual and group learning. The workshop website is http://agileviz.net.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

The Design of AIED Systems: Learner-Centred and Design Pattern Approaches

Vladan DEVEDZIC [a], Judith GOOD [b], Andreas HARRER [c], Rose LUCKIN [d], Darren PEARCE [e], Symeon RETALIS [f] and Judy ROBERTSON [g]
[a] *University of Belgrade, devedzic@fon.bg.ac.yu*
[b] *University of Sussex, J.Good@sussex.ac.uk*
[c] *University of Duisburg-Essen, harrer@collide.info*
[d] *London Knowledge Lab, r.luckin@ioe.ac.uk ,*
[e] *London Knowledge Lab, d.pearce@ioe.ac.uk*
[f] *University of Piraeus, retal@unipi.gr*
[g] *Heriot-Watt University, Judy.Robertson@hw.ac.uk*

Learner-Centred Design (LCD) is a methodology used in the development of educational systems which involves the learner and any other relevant stakeholders in the design process. Stakeholders such as learners, parents and teachers may be involved in this process as fully-fledged design partners, as a source of information which is fed into the design or as a tester of prototypical systems. This is invaluable for requirements gathering through observation of current practice, critiquing of existing systems, low-tech prototyping and formative evaluation of high-tech prototypes.

A related, but distinctly different area, is Design Patterns. In general, such patterns make explicit sound practices of how to solve repeatedly-occurring design problems and provide re-usable knowledge for creating systems. Since the origin of design patterns in the area of architecture in 1977, analogous approaches have been developed in a variety of domains. In particular, in the last decade, design patterns have been used extensively in software engineering and, more specifically, have recently been applied to human-computer interaction, groupware and software development.

Since both techniques draw on the identification of guidelines of how to design systems, this workshop aims to explore the potential synergy between these two approaches specifically in the context of the design of AIED systems. The workshop will consist of a series of presented papers in these areas. Contributors will also be asked to provide accompanying posters of their work to facilitate discussion throughout the day. The workshop will conclude with a plenary session led by an expert panel drawing together the effective techniques that have been running themes within the workshop presentations.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Educational Data Mining Workshop

Cecily HEINER [a], Neil HEFFERMAN [b] and Tiffany BARNES [c]
[a] *University of Utah*, *cecily@cs.utah.edu*
[b] *Worcester Polytechnic Institute*, *nth@wpi.edu*
[c] *University of North Carolina at Charlotte*, *tbarnes2@uncc.edu*
http://www.educationaldatamining.org/

Introduction

Educational data mining is the process of converting raw data from educational systems to useful information that can be used to inform design decisions and answer research questions. Data mining encompasses a wide range of research techniques that includes more traditional options such as database queries and simple automatic logging as well as more recent developments in machine learning and language technology.

Educational data mining techniques are now being used in ITS and AIED research worldwide. For example, researchers have used educational data mining to:

- Detect affect and disengagement
- Detect attempts to circumvent learning called "gaming the system"
- Guide student learning efforts
- Develop or refine student models
- Measure the effect of individual interventions
- Improved teaching support
- Predict student performance and behavior

However, these techniques could achieve greater use and bring wider benefits to the ITS and AIED communities. We need to develop standard data formats, so that researchers can more easily share data and conduct meta-analysis across tutoring systems, and we need to determine which data mining techniques are most appropriate for the specific features of educational data, and how these techniques can be used on a wide scale. The workshop will provide a forum to present preliminary but promising results that can advance our knowledge of how to appropriately conduct educational data mining and extend the field in new directions.

Topics

They include, but are not limited to:

- What new discoveries does data mining enable?
- What techniques are especially useful for data mining?
- How can we integrate data mining and existing educational theories?
- How can data mining improve teacher support?
- How can data mining build better student models?
- How can data mining dynamically alter instruction more effectively?
- How can data mining improve systems and interventions evaluations?
- How do these evaluations lead to system and intervention improvements?
- How is data mining fundamentally different from other research methods?

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Emerging Technologies for Inquiry-Based Learning in Science

Dawn WOODGATE and Danaë STANTON FRASER
Department of Psychology, University of Bath, UK

Abstract. The aim of this workshop is to promote discussion and collaboration among educational practitioners and researchers, and others with an interest in using new technologies to stimulate learners' interest and motivation in science topics, and to explore ways in which emerging technologies can be appropriated for use in scientific investigations in schools and elsewhere, to bring about a more engaging and hands-on approach to science learning.

Keywords. Science education, inquiry-based learning, emerging technologies

Introduction

Concerns have been expressed about a perceived lack of interest in science among both school students and the general public. Quite apart from any economic concerns, science forms an important part of our culture, and therefore a measure of scientific understanding is necessary if people are not to be largely excluded from participating in many topical debates. However, exactly how we can enthuse (or re-enthuse) people with science remains largely unclear. One way might be through adopting different approaches to science education. We aim to explore a broad range of questions related to this topic, some of which may cut across disciplinary boundaries, by conceptualizing the terms 'science education' and 'educational establishments' in the broadest possible light, to include what is sometimes called 'informal' learning, alongside traditional classroom based education of the type that takes place in schools, colleges and universities. 'Informal learning' is perhaps a somewhat controversial term, since no clear boundary exists between 'formal' and 'informal' learning, but we take it here to mean learning that takes place outside the classroom, including the types of activities that are referred to as 'public engagement' as opposed to 'education'.

We invite the participation of those interested in a broad range of topics including, but not limited to: which technologies? eg, simulations, mixed/augmented reality, embedded and mobile systems, eScience / distributed, ubiquitous/pervasive technologies for learning; theoretical issues, e.g., from education, psychology or sociology, as related to development and use of emerging technologies; case studies in use of emerging technologies in science education; issues of integrating emerging technologies into the classroom context; emerging technologies for learning in science in schools, colleges and universities and / or 'informal' learning contexts; emerging technologies and science curricula; problems and issues of scaling –up pilot projects; emerging technologies, science education and the potential for wider public engagement in science.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

From Communities of Interest to Social Learning Communities

Beatriz BARROS [a] and Julita VASSILEVA [b]
[a] *UNED, Spain, bbarros@lsi.uned.es*
[b] *University of Saskatchewan, Canada, jiv@cs.usask.ca*

Web technologies have opened a number of possibilities for group formation, communication, sharing and learning. These technologies can support a variety of forms of learning within web-based learning communities, by proposing activities that involve collaboration and communication. Online communities of interest emerge, where the "interest" lies in group learning and can be supported by rich and varied collaborative processes. We call this emerging area of research "Social Learning Communities" and distinguish them from the areas of Online or Virtual Communities by emphasizing the social aspects of learning, models for representing and capturing collective knowledge, group profiles, computer-supported collaborative workspaces, distributed management systems, facilities for sharing searching and reusing.

The goal of this workshop is to provide a forum for the discussion of recent trends and perspectives in Social Learning Communities, focusing on Artificial Intelligence models, techniques and procedures to assist learners, to model better the teachers' activities, to follow the learning processes and, in short, to improve the learners' experience and results.

The list of topics includes, but is not limited to: Instructional Methods for Social Learning Communities (SLC); Motivation in Social Learning; Methods and Mechanisms for sharing and reusing data in SLC; Collaboration Schemas and Scripts; Approaches to Scaffolding Collaboration; Collaborative Learning in Virtual Communities; Communications Modes; Group Modelling in SLC; Analysis and Monitoring in SLC; Intelligent Methods for Regulating SLC; Dynamic Group Formation; Methodologies for Building Collective Knowledge; Theories for SLC; Collaborative Filtering and Recommender systems; Motivation in Social Learning

Workshop chairs
Beatriz Barros and Julita Vassileva

Program Committee
Harrer Andreas, Nikolaus Avouris, Nelson Baloian, Philip Bonanno, Rosa M. Carro Weiqin Chen, Rosta Farzan, Elena Gaudioso, Ulrich Hoppe, Piet Kommers, Marcelo Milrad, Niels Pinkwart, Demetrios Sampson, Margriet Simmerling, Amy Soller, Felisa Verdejo

Modeling and Scaffolding Affective Experiences to Impact Learning

Sidney D'MELLO [a], Scotty CRAIG [b], Rana el KALIOUBY [c], Madeline ALSMEYER [d] and Genaro REBOLLEDO-MENDEZ [d]
[a] *University of Memphis,* [b] *University of Pittsburgh ,*
[c] *Massachusetts Institute of Technology,* [d] *University of Sussex*

Learning is accompanied by episodes of success and failure that inevitably invoke a host of associated affective responses. Interest driven content exploration (curiosity), being encouraged by success (happiness), making mistakes (feeling confused), recovering from them (overcoming frustration), diagnosing what went wrong (not becoming dispirited), and starting over again (with hope, determination, and maybe even enthusiasm) are the natural phases involved in the mastery of deep-level concepts. While, the last decade has been ripe with research investigating the interplay between emotions and learning, there are several open challenges hindering progress in this area. These include *empirical and theoretical questions* such as: (1) What are the emotions that are important to learning? (2) How are they linked with cognition and metacognitive processes (e.g. self-regulation and goal orientation)? (3) How do they get recognized by tutors, peers, and the learners themselves?

The next generation of educational technologies needs to be more than mere cognitive machines. Their educational strategies should be tailored in order to restore the balance between cognition and affect. This transition into the affective domain requires innovative approaches to construct online models of the emotion dynamics of a learner and efficiently utilize these models to optimize learning. In addition to the questions raised above, this endeavor brings to light additional *computational issues:* (1) How can we develop and evaluate systems to automatically detect learner centric emotions in real-time? (2) How should intelligent learning environments modify their dialogue planners to be responsive and reactive to the learner's affect? (3) What social rules should our embodied conversational agents that serve as artificial tutors or peer learning companions employ in order to synthesize affective expressions so as to yield more naturalistic communication?

This multidisciplinary area within the learning sciences includes researchers from psychology, education, cognitive science, computer science, artificial intelligence, and neuroscience. Since many of these researchers share the same goals of developing learning environments that effectively coordinate pedagogy with the learner's emotions, the proposed workshop seeks to create crosstalk between the areas. By providing a framework to discuss and evaluate novel research we hope to leverage recent advances to speed-up future research in this area.

We recognize the need for broad, fleshed-out, theories of affect and learning that can be implemented in computational architectures. While we do not expect to construct a theory from this workshop alone, we hope that merging perspectives from the learning sciences with recent advances in artificial intelligence will help set the foundation for a comprehensive theory to scaffold future research in this area.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Narrative Learning Environments

Mark RIEDL [a], Isabel Machado ALEXANDRE [b], Brian MAGERKO [c], Ana PAIVA [d], and H. Chad LANE [a]
[a] *University of Southern California, Institute for Creative Technologies*
[b] *Instituto Superior de Ciências do Trabalho e da Empresa*
[c] *Michigan State University*
[d] *Instituto de Engenharia de Sistemas e Computadores Investigação e Desenvolvimento*

Narrative and its closely related counterpart, storytelling, are most often thought of as modalities for entertainment. Regardless, narrative has long been a part of the repertoire of tools employed by instructors, trainers, and educators to achieve their pedagogical goals. Psychologists have recognized narrative as relevant to the way we store and make sense of episodic experience, often described as the phenomenon of narrative construction of reality. Recent technologies, especially pertaining to artificial intelligence, have demonstrated intelligent learning environments that incorporate narrative in the form of "Edutainment" and "Serious Games," which blend computer game play for more engaging and distributable learning and training.

The goal of this workshop is to bring together researchers from different disciplines to discuss the role of narrative in education and training and to report on techniques, tools, and strategies for the creation of what we call Narrative Learning Environments (NLEs). In our view, a Narrative Learning Environment is a type of Intelligent Learning Environment in which narrative is a key instrument in facilitating teaching and learning. We intend to shed light on the possible roles that narrative can play in intelligent learning environments, techniques for using narrative to educate and train, and tools for building Narrative Learning Environments.

The topics and research questions we intend to address are as follows:

- What are the best knowledge representations for narrative in NLEs?
- How can existing methods be adapted to improve the quality of "designed in" story?
- What are the implications of using a narrative-centered approach in a learning environment?
- What theories of interactive narrative are needed to support the different possibilities provided by different delivery mechanisms?
- How can we design authoring tools to support the development, scripting, or generation of narratives for Narrative Learning Environments?
- What can we learn from game designers that can be used within NLEs?
- How can notions of interactive narrative support the development of relationships between learners and teachers?
- How does narrative contribute to the effectiveness of learning environments? For what kinds of tasks are narrative approaches *not* ideal?
- What tradeoffs are there between pedagogy and narrative?
- How does one evaluate Narrative Learning Environments?

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

SWEL'07: International Workshop on Ontologies and Semantic Web Services for Intelligent Distributed Educational Systems

Nicola CAPUANO[a], Darina DICHEVA[b], Andreas HARRER[c], Riichiro MIZOGUCHI[d]
[a] *University of Salerno, Italy, capuano@crmpa.unisa.it,*
[b] *Winston-Salem State University, USA, dichevad@wssu.edu*
[c] *University of Duisburg-Essen, Germany, harrer@collide.info*
[d] *University of Osaka, Japan, miz@ei.sanken.osaka-u.ac.jp*
SWEL web site: http://compsci.wssu.edu/iis/swel/swel07/swel07-aied07.html

The SWEL'07 workshop at AIED'07 aims to address issues related to using Semantic Web based knowledge representation and Grid technologies for the design and development of intelligent distributed educational systems. It covers topics related to the use of ontologies, Semantic Web standards, and distributed processes enabled by Grid technologies for learning content delivery, services and knowledge components specification, effective intelligent courseware construction, and learner modelling.

SWEL'07 is the fifth in a series, following the successful edition in 2002, held in conjunction with ICCE'02; the three sessions of SWEL'04 in conjunction with ITS'04, AH'04, and ISWC'04; the three sessions of SWEL'05 in conjunction with AIED'05, ICALT'05, and K-CAP'05; and SWEL'06 held in conjunction with AH'06.

This edition has a special focus on service-oriented Semantic Grid Architectures for e-Learning and is organized together with two SIGs of the European Network of Excellence Kaleidoscope "Artificial Intelligence and Education" and "Learning Grid".

Two sessions are included in the workshop, focused on the following topics:

1. *Session on Ontologies and Semantic Web Standards for e-Learning*
   - Building ontologies for e-learning; theoretical issues in ontology engineering.
   - Using ontologies and Semantic Web standards for structuring, representing, indexing, and retrieving shareable and interoperable learning resources.
   - Using ontologies and SW standards for supporting authoring of intelligent educational systems.
   - Using SW-based contexts for personalization of e-learning applications.
2. *Session on Semantic Grid Architectures and e-Learning Services*
   - Semantic Grid architectures and systems for distributed intelligent e-learning.
   - Models and tools for the semantic annotation of e-learning services.
   - Automatic composition of semantically described services and processes.
   - Pedagogical approaches and learning models for exploiting the potential of distributed and Grid technologies for e-learning.
   - Application scenarios for dynamic composition of distributed learning services using AIED and GRID technologies and techniques.

The workshop also features a keynote talk "Inside a Theory-Aware and Standards-Compliant Authoring System" by Riichiro Mizoguchi.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Workshop on Metacognition and Self-Regulated Learning in ITSs

Ido ROLL [a], Vincent ALEVEN [a], Roger AZEVEDO [b], Ryan S.J.D. BAKER [c], Gautam BISWAS [d], Cristina CONATI [e], Amanda HARRIS [f], Rose LUCKIN [g], Antonija MITROVIC [h], Tom MURRAY [i], Barbara WHITE [j], Phil WINNE [k]

[a] *Carnegie Mellon University,* [b] *University of Memphis,* [c] *University of Nottingham,* [d] *Vanderbilt University,* [e] *University of British Columbia,* [f] *University of Sussex,* [g] *London Knowledge Lab,* [h] *University of Canterbury,* [i] *University of Massachusetts, Amherst,* [j] *University of California, Berkeley,* [k] *Simon Fraser University*

Introduction

A number of studies in educational psychology and cognitive science have shown that students who apply better metacognitive skills and self-regulation strategies during their learning process have better learning outcomes. Educational technology such as intelligent tutoring systems has proven to be effective at the domain (or cognitive) level but similar success has not been achieved yet with regard to tutoring better metacognitive and self-regulation skills. A key question is whether instructional technology can be as effective in fostering metacognitive skills as it is in teaching domain-specific skills and knowledge. On the face of it, the answer is positive. Novel means for interaction, better understanding of learning, mechanisms for tracing students' knowledge, and established domain-level tutoring principles could be applied at the metacognitive level. However, it remains largely unknown exactly how educational technology can help students acquire better metacognitive skills and use them more effectively.

The aim of the workshop is to improve our understanding of the design of *goals*, *instruction*, and *assessment* of tutoring metacognition and self-regulated learning using educational technology.

Topics of interest

Topics of interest include, but are not limited, to:

- Goals for metacognitive tutoring
- Design guidelines for metacognitive tutors
- Assessment metacognitive knowledge and learning
- Integration of metacognitive and cognitive instruction
- Metacognitive models and representation of metacognitive knowledge
- Pedagogies to teach metacognition
- Empirical or descriptive studies evaluating metacognitive tutoring
- Metacognition and motivation
- Metacognitive awareness of Intelligent Tutoring Systems

Tutorial Abstracts

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Authoring Constraint-Based Tutoring Systems

Antonija MITROVIC [a], Stellan OHLSSON [b], Brent MARTIN [a] and Pramuditha SURAWEERA [a]
[a] *Department of Computer Science and Software Engineering University of Canterbury, New Zealand*
[b] *Department of Psychology, University of Illinois at Chicago*

This tutorial covers both the theory and practice of Constraint-Based Modeling (CBM). The first part of the tutorial introduces constraints and CBM as a theoretical foundation for Intelligent Tutoring Systems (ITSs). The second part covers ASPIRE, our new authoring system for developing constraint-based ITSs, and gives participants the opportunity to experience developing a small ITS in ASPIRE.

We will first introduce constraints as a way of representing domain knowledge. Currently, cognitive models typically cast declarative knowledge as consisting of propositions – knowledge units that encode assertions (which can be true or false) that support description, deduction and prediction. We have developed an alternative model of declarative knowledge that consists of constraints – units of knowledge that are more prescriptive than descriptive, and that primarily support evaluation and judgment. We present a formal representation of constraints and explain its conceptual rationale, and then introduce two applications of CBM. The first is the use of constraints as a basis for a machine learning algorithm that allows a heuristic search system to detect and correct its own errors. From this point of view, constraint-based learning is a form of adaptive search. This algorithm was originally developed as a hypothesis about how people learn from errors. We present the algorithm in some detail and briefly summarize applications to various problems in the psychology of cognitive skill acquisition.

Next we develop in detail the application of CBM to the design and implementation of ITSs. The constraint-based knowledge representation provides a novel way to represent the target subject matter knowledge, which has the advantage of directly supporting one of the main functions of expert knowledge in an ITS: to detect student errors. More importantly, the constraint-based representation provides a theoretically sound and practical solution to the intractable problem of student modeling. Finally, constraints and the associated learning algorithm provide detailed implications for how to formulate individual tutoring messages. We present multiple systems that follow this blueprint, together with empirical evaluation data.

In the second part of the tutorial we present our new authoring system (ASPIRE). The goal of ASPIRE is to make ITS authoring available to educators who have little technical knowledge of ITSs or programming in general. ASPIRE does this by providing extensive authoring support, such that the author, as far as possible, is always working at the domain knowledge level, not the programming level. We describe its architecture and functionality, as well as the authoring procedure it supports. The participants will then have hands-on opportunities to investigate ASPIRE closer, by using it to build a simple tutor.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Developing Agent-Based AIEd Systems under the Perspective of the Semantic Web

Evandro COSTA [a,1] and Ig BITTENCOURT [a,b,2]
[a] *Computer Science Institute -Federal University of Alagoas*
GrOW -Group of Optimization of the Web
[b] *Federal University of Campina Grande, Brazil, Paraíba*

Introduction

AIED Systems are gradually incorporating Semantic Web technology. The general motivation for this movement is to making the Web more understandable by machines and then generating AIED systems more adaptive and intelligent. This tutorial aims at providing a systematic study of constructing AIED systems under the perspective of Semantic Web, bringing together theory and practice. It will firstly provide the participants with an understanding of the semantic web concepts and technologies. Additionally, the participants will be taught: i) how to evaluate the usage of agents and semantic web technologies and ii) how to implement these technologies in the context of AIED systems. Then, some of the topics to be covered during the tutorial include: agent technologies (with applications, and standards from the semantic web), main components in AIED systems (domain, student, pedagogical, and collaborative) using semantic web ontology language, methods and methodologies for building agent-based AIED systems under the perspective of the semantic web, and semantic web services.

Audience

The tutorial welcomes researchers that are interested in exploring the field of semantic web in the context of AIEd systems. In addition, the tutorial welcomes developers who are interested in implementing AIEd systems involving semantic web by adopting agent technology. It is assumed that participants have some basic artificial intelligence and software engineering knowledge.

[1] Corresponding Author: Federal University of Alagoas, Brazil, Alagoas, Campus A. C. Simões, BR 104 - Norte, Km 97, Cidade Universitária - Maceió - AL, CEP 57072-970; E-mail: evandro@ic.ufal.br or ebc@fapeal.br
[2] Corresponding Author: Federal University of Campina Grande, Brazil, Paraíba, Rua Aprígio Veloso, 882 - Bodocongó, CEP 58.109-900. E-mail: ibert@dsc.ufcg.edu.br and ibert@ic.ufal.br

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Mediating Conceptual Knowledge with Qualitative Reasoning Technology

Bert BREDEWEG [a,1] and Paulo SALLES [b]
[a] *Human Computer Studies, University of Amsterdam, The Netherlands*
[b] *Institute of Biological Sciences, University of Brasília, Brazil*

Introduction

Qualitative Reasoning is a research area within Artificial Intelligence that focuses on means to articulate and communicate conceptual knowledge such as system structure, causality, the start and end of processes, the assumptions and conditions under which facts are true, qualitative distinct behaviours, etc. (cf. [2,3]). Humans continuously reason about the physical world. Understanding how systems behaved in the past and how they may behave in the future is a crucial aspect of human nature. Qualitative Reasoning investigates how this aspect of human intelligence can be automated on computers.

Recently Garp3 has been developed, a user-friendly workbench that allows modellers to build, simulate, and inspect qualitative models [1]. The Garp3 software is available via www.garp3.org and is developed as part of the NaturNet-Redime project (www.naturnet.org), which develops educational materials for teaching and training on issues relevant to Sustainable Development. Garp3 can be used in educational settings to have learners create models and thus have them 'learn by modelling'. Alternatively, learners can be given ready-made models and work through a set of assignments while interacting with models and their simulation results (simulation-based learning). Both approaches can be administered using an individual or a collaborative approach to learning. The software provides additional functionality to support collaboration, such as the ability to share and re-use models (and model parts) via an online repository.

Conceptual models are important instruments for science education. With the support of the Garp3 workbench a wide range of possibilities for educational research and applications have emerged. The tutorial addresses these possibilities.

References

[1] Bredeweg, B., Bouwer, A., Jellema, J., Bertels, D., Linnebank, F. and Liem, J. Garp3 – A new Workbench for Qualitative Reasoning and Modelling. 20th International Workshop on Qualitative Reasoning (QR-06), C. Bailey-Kellogg and B. Kuipers (eds), pages 21-28, Hanover, New Hampshire, USA, 10-12 July, 2006.
[2] Bredeweg B. and Struss P. (eds). Current Topics in Qualitative Reasoning. AI Magazine (special issue), 24(4): 13-130, 2003.
[3] Weld, D. & de Kleer, J. (eds.) Readings in Qualitative Reasoning about Physical Systems. San Mateo, CA: Morgan Kaufmann, 1990.

[1] Corresponding Author: Kruislaan 419, 1098 VA Amsterdam, The Netherlands, bredeweg@science.uva.nl

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Rapidly Creating a Tutorial Dialogue System Using the TuTalk Tool Suite

Pamela W. JORDAN and Kurt VANLEHN
Learning Research and Development Center
University of Pittsburgh, Pittsburgh, Pennsylvania, USA

Most tutorial dialogue systems that have undergone successful evaluations represent development efforts of many man-years (e.g. CIRCSIM, AutoTutor, WHY-Atlas, the Geometry Explanation Tutor, SCoT, inter alia). Several projects have developed methodologies and tools to improve the effectiveness and development costs of computer-based tutorial dialogue.

This tutorial will begin with an overview of various dialogue authoring methodologies and the lessons learned in creating and delivering dialogue content. We will then prepare participants to create a small dialogue system of their own using the TuTalk tool suite. We will introduce them to this tool suite and demonstrate how to use it to build a small dialogue system. Participants will then break into groups and develop a small dialogue system of their own. Participants can either expand on the simple domain we will use in our demonstration or they may select one of their own.

We encourage all participants to bring their laptops to the tutorial and to install the TuTalk authoring component prior to the tutorial. Specific instructions and help will be made available on our website in the weeks prior to the tutorial (see http:andes3.lrdc.pitt.edu/TuTalk). TuTalk is a community resource that is freely available to researchers. It builds upon our prior work in developing basic dialogue technology and tools for the rapid development of running dialogue systems and has been used to build over 14 successful dialogue systems that have been used in experiments with students.

Audience

The intended audience includes researchers who study tutorial dialogue, regardless of their expertise with software technology and tutorial systems researchers who wish to add a natural language dialogue capability to their systems.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

TagHelper Tools: Tools for Supporting the Analysis of Verbal Data

Carolyn P. ROSÉ
Language Technologies Institute & Human-Computer Interaction Institute
Carnegie Mellon University, Pittsburgh, PA 15213
http://www.cs.cmu.edu/~cprose/TagHelper.htm
cprose@cs.cmu.edu

In this tutorial, we present a publicly available tool called TagHelper tools that can be used to automate the analysis of conversational data using machine learning technology. Conversational process analysis is especially important in the areas of Computer Supported Collaborative Learning and Tutorial Dialogue research. Thus, one important goal of our research is to provide valuable tools to support research in these areas. The same technology can also be used to perform on-line assessments of learning processes during collaborative learning or tutoring conversations, which may then be used to trigger context appropriate interventions or provide reports to group learning facilitators.

TagHelper's basic classification functionality is now available to researchers to use in their own work. Because we have observed the popular usage of Microsoft Excel as an environment in which researchers commonly do their corpus analysis work, we have adopted that as our standard file format. The TagHelper application and documentation can now be downloaded free of charge from the TagHelper tools website at http://www.cs.cmu.edu/~cprose/TagHelper.html. Note that this version of TagHelper that is publicly available is designed to work with pre-segmented English, German, or Chinese texts, although additional licensing requirements are associated with the Chinese functionality through Academia Sinica in Taiwan. In order to make TagHelper tools accessible to the widest possible user base, we have set up its default behavior in such a way that users are only required to provide examples of annotated data along with un-annotated data. At the click of a button, TagHelper tools builds a model based on the annotated examples that it can then apply to the un-annotated examples. More advanced users can experiment with a variety of available settings, which may allow them to achieve better performance with TagHelper tools.

The tutorial will begin with an overview of the various ways in which TagHelper tools can be used to support educational technology research. We will then give a tour of TagHelper tools' basic functionality. The focus of the bulk of the tutorial will be on strategies for getting the best performance out of TagHelper tools. We will also provide attendees with pointers to resources for more advanced training.

We invite tutorial attendees to bring their own conversational data with them to work with. We will also provide sample data sets for attendees to work with. We also strongly suggest that attendees bring their own laptops, preferably with the software already installed. A new version of TagHelper tools will be released on the TagHelper tools website the week prior to the tutorial.

This page intentionally left blank

Interactive Events Summaries

This page intentionally left blank

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Interactive Event: Ontological Domain Modeling of Constraint-Based ITS

Brent MARTIN, Antonija MITROVIC and Pramuditha SURAWEERA
Intelligent Computer Tutoring Group
University of Canterbury, Christchurch, New Zealand
{brent.martin, tanja.mitrovic, pramudi.suraweera}@canterbury.ac.nz

1. Overview

The WETAS ITS shell supports the authoring of ITS for constraint-based tutors by providing an engine that performs the main functions of a tutor (interface, account/security handling, student modeling, pedagogical functions). In particular, for text-based tutors the user only provides the domain model and problem set; WETAS does the rest. WETAS-Ontology is a graphical ontology editor that allows authors to develop the domain ontology, from which the domain model for WETAS is then automatically generated. WETAS-Ontology was successfully trialled in June 2006 at the AH2006 e-learning summer school, National College of Ireland, Dublin.

Participants will use WETAS-Ontology to develop the domain model for a simple tutoring system, in a domain chosen from three that will be provided. The domains will be such that the majority of participants would be expected to complete a fairly simple ITS in the time available, while more advanced participants should be able to produce more comprehensive systems. Participants will be scaffolded such that they only need to produce the ontology; the other functions normally performed by an author (writing the problem statements in particular) will have been completed by the event organizers. The participants will then be able to trial each other's tutors. The session will end with a discussion of everyone's experiences.

2. WETAS-Ontology

WETAS-Ontology allows the user to specify the domain diagrammatically using ontology. Nodes in the diagram specify the objects of the domain. These contain slots for the object name, how it is identified in a solution, and any constraints upon this object (such as a maximum cardinality). Arcs represent relationships, specifying constraints upon combinations of objects. WETAS-Ontology supports (and reasons about) three types of relationships: *isa* (concept A is a specialisation of concept B), *part-of* (concept A is a part of concept B) and *equivalent instance of* (concepts A and B are equivalent examples of concept C). The system then uses templates (or "meta-constraints") to generate a set of constraints for the domain from the nodes in the ontology, which can be run by WETAS. Constraints can be edited manually or added to if required. Authors are also free to develop constraint "macros" that can be used by the generated constraints to specify more complex semantics.

Artificial Intelligence in Education
R. Luckin et al. (Eds.)
IOS Press, 2007

Tactical Language and Culture Training System: Learn and Author

W. Lewis JOHNSON
Alelo, Inc. and USC / ISI
11965 Venice Bl., Suite 402, Los Angeles, CA 90066
ljohnson@alelo.com; johnson@isi.edu

Abstract. The Tactical Language and Culture Training System (TLCTS) is an AI-enhanced learning environment platform that helps learners quickly acquire communication skills in foreign languages and cultures. It integrates intelligent tutoring and serious game capabilities into a package that helps learners quickly acquire communication skills. Thousands of copies of TLCTS-based learning systems are currently distributed around the world. This Interactive Event will give participants an opportunity to experience learning with TLCTS, and an introduction to content authoring for TLCTS.

Keywords. Game-based learning, second language learning, pedagogical agents, authoring

The Tactical Language and Culture Training System (TLCTS) is an AI-enhanced learning environment platform that helps learners quickly acquire communication skills in foreign languages and cultures. It has been developed by Alelo, Inc. based on a prototype developed at the University of Southern California (USC). It utilizes an integrated combination of intelligent tutoring system and serious game technologies. Trainees work through a series of interactive lessons and exercises, called the Skill Builder, focusing on mission-oriented communication skills. The lessons make extensive use of automated speech recognition focused on learner language, and provide learners with feedback on their performance. Cultural notes describing customs and nonverbal gestures are integrated into the Skill Builder lessons. Trainees apply their skills in an interactive Arcade Game, where they use spoken commands in the target language to navigate a town grid and destroy enemies, and in a Mission Game, where they must interact with simulated local people in order to accomplish their mission. Three TLCTS training courses have been developed so far: Tactical Levantine™, focusing on Levantine Arabic, Tactical Iraqi™, focusing on Iraqi Arabic, and Tactical Pashto™, focusing on the Pashto dialect spoken in Southern Afghanistan. Tactical French courses are under development. Thousands of US military users have trained with the system, and consistently rate it highly. It will shortly be made available to service members in allied military forces, as well as the general public.

The approach is not specific to the military. The key is that the courses are task-oriented: the learner has a task to carry out, the Skill Builder helps the learner to acquire the skills necessary to carry out task, and the Mission Game gives the learner opportunity to practice the task in simulated settings.

The interactive event will consist of the following parts. First, the participants will have an opportunity to work with a TLCTS course in depth. Then, participants will have the opportunity to propose modifications to an existing TLCTS course; the presenter will then author the modifications on the spot.

Author Index

This page intentionally left blank

This page intentionally left blank